SERiF

v0.0.1a

Generated by Doxygen 1.13.2

1 SERiF 1

2 Todo List 3

3 Namespace Index 5
3.1 Namespace List . . . 5

4 Concept Index 7
4.1 Concepts . . . 7

5 Hierarchical Index 13
5.1 Class Hierarchy . . . 13

6 Class Index 27
6.1 Class List . . . 27

7 File Index 49
7.1 File List . . . 49

8 Namespace Documentation 57
8.1 mfem Namespace Reference . . . 57
8.2 serif Namespace Reference . . . 57
8.3 serif::dimensions Namespace Reference . . . 57
8.3.1 Typedef Documentation . . . 62
8.3.1.1 AccelerationValue . . . 62
8.3.1.2 AngularMomentumValue . . . 62
8.3.1.3 AngularVelocityValue . . . 62
8.3.1.4 AreaValue . . . 62
8.3.1.5 ChemicalPotentialValue . . . 62
8.3.1.6 DensityValue . . . 62
8.3.1.7 DimensionlessValue . . . 62
8.3.1.8 DynamicViscosityValue . . . 63
8.3.1.9 EnergyDensityValue . . . 63
8.3.1.10 EnergyValue . . . 63
8.3.1.11 EntropyValue . . . 63
8.3.1.12 ForceValue . . . 63
8.3.1.13 FrequencyValue . . . 63
8.3.1.14 GravitationalPotentialValue . . . 63
8.3.1.15 InternalEnergyValue . . . 64
8.3.1.16 KinematicViscosityValue . . . 64
8.3.1.17 LengthValue . . . 64
8.3.1.18 LuminosityValue . . . 64
8.3.1.19 MagneticFluxDensityValue . . . 64

8.3.1.20 MassFlowRateValue . . . 64
8.3.1.21 MassValue . . . 64
8.3.1.22 MomentOfInertiaValue . . . 65
8.3.1.23 MomentumValue . . . 65
8.3.1.24 NeitherConstNorVolatile . . . 65
8.3.1.25 NeitherConstNorVolatileReference . . . 65
8.3.1.26 NumberDensityValue . . . 65
8.3.1.27 OpacityValue . . . 65
8.3.1.28 PowerValue . . . 65
8.3.1.29 PressureValue . . . 66
8.3.1.30 QuantityOfT . . . 66
8.3.1.31 SpecificEnergyValue . . . 66
8.3.1.32 SpecificEnthalpyValue . . . 66
8.3.1.33 SpecificEntropyValue . . . 66
8.3.1.34 SpecificInternalEnergyValue . . . 66
8.3.1.35 TemperatureValue . . . 66
8.3.1.36 TimeValue . . . 67
8.3.1.37 TorqueValue . . . 67
8.3.1.38 VelocityValue . . . 67
8.3.1.39 VolumeValue . . . 67
8.4 serif::dimensions::runtime Namespace Reference . . . 67
8.4.1 Variable Documentation . . . 68
8.4.1.1 thermodynamicQuantityID . . . 68
8.5 serif::discretization Namespace Reference . . . 68
8.6 serif::discretization::backend Namespace Reference . . . 68
8.7 serif::discretization::backend::mfem Namespace Reference . . . 69
8.7.1 Enumeration Type Documentation . . . 70
8.7.1.1 ContractionMode . . . 70
8.7.1.2 ReferenceOperation . . . 70
8.7.2 Function Documentation . . . 70
8.7.2.1 select_boundary_dofs() . . . 70
8.7.2.2 select_point_dofs() . . . 71
8.8 serif::discretization::blocks Namespace Reference . . . 72
8.8.1 Typedef Documentation . . . 75
8.8.1.1 FieldOf . . . 75
8.8.1.2 FunctionalSpaceOf . . . 75
8.8.1.3 QuantityOf . . . 75
8.8.1.4 SupportOf . . . 76
8.8.1.5 TermsOf . . . 76
8.8.2 Variable Documentation . . . 76

8.8.2.1 DYNAMIC_BLOCK_SIZE . . . 76
8.8.2.2 field_descriptor . . . 76
8.8.2.3 term_descriptor . . . 77
8.9 serif::discretization::domain Namespace Reference . . . 77
8.9.1 Typedef Documentation . . . 78
8.9.1.1 AllDomains . . . 78
8.9.1.2 StellarDomains . . . 78
8.9.2 Function Documentation . . . 78
8.9.2.1 CountVariadicArguments() . . . 78
8.9.3 Variable Documentation . . . 79
8.9.3.1 AIsBaseClassOfB . . . 79
8.9.3.2 is_domain_set_v . . . 79
8.9.3.3 is_domain_set_v< DomainSet< DomainTs... > > . . . 79
8.10 serif::discretization::domain::ids Namespace Reference . . . 79
8.10.1 Variable Documentation . . . 80
8.10.1.1 is_boundary_id_v . . . 80
8.10.1.2 is_boundary_id_v< BoundaryID< BoundaryT, IdentifierValue > > . . . 80
8.10.1.3 is_domain_id_v . . . 81
8.10.1.4 is_domain_id_v< DomainID< DomainT, IdentifierValue > > . . . 81
8.11 serif::discretization::domain::ids::lists Namespace Reference . . . 81
8.11.1 Function Documentation . . . 82
8.11.1.1 CountVariadicArguments() . . . 82
8.11.2 Variable Documentation . . . 82
8.11.2.1 is_boundary_id_list_v . . . 82
8.11.2.2 is_boundary_id_list_v< BoundaryIDList< BoundaryIDTs... > > . . . 82
8.11.2.3 is_domain_id_list_v . . . 83
8.11.2.4 is_domain_id_list_v< DomainIDList< DomainIDTs... > > . . . 83
8.12 serif::discretization::domain::ids::validation Namespace Reference . . . 83
8.12.1 Function Documentation . . . 84
8.12.1.1 all_unique_values() . . . 84
8.13 serif::discretization::domain::mesh Namespace Reference . . . 84
8.14 serif::discretization::domain::relation Namespace Reference . . . 84
8.14.1 Function Documentation . . . 85
8.14.1.1 CountVariadicArguments() . . . 85
8.14.2 Variable Documentation . . . 85
8.14.2.1 AIsBaseClassOfB . . . 85
8.15 serif::discretization::domain::relation::lists Namespace Reference . . . 85
8.15.1 Function Documentation . . . 86
8.15.1.1 CountVariadicArguments() . . . 86
8.15.2 Variable Documentation . . . 86

8.15.2.1 is_relation_list_v . . . 86
8.15.2.2 is_relation_list_v< RelationList< RelationTs... > > . . . 86
8.16 serif::discretization::domain::relation::validation Namespace Reference . . . 86
8.16.1 Enumeration Type Documentation . . . 87
8.16.1.1 RelationValidationFailure . . . 87
8.17 serif::discretization::domain::schema Namespace Reference . . . 87
8.17.1 Typedef Documentation . . . 89
8.17.1.1 CoreEnvelopeVacuumDomainSchema . . . 89
8.17.2 Function Documentation . . . 89
8.17.2.1 make_attribute_marker() . . . 89
8.17.3 Variable Documentation . . . 89
8.17.3.1 is_schema_v . . . 89
8.17.3.2 is_schema_v< DomainSchema< DomainIDs, BoundaryIDs, Relations > > . . . 89
8.18 serif::discretization::domain::schema::validation Namespace Reference . . . 90
8.18.1 Enumeration Type Documentation . . . 90
8.18.1.1 RelationValidationFailure [1/3] . . . 90
8.18.1.2 RelationValidationFailure [2/3] . . . 90
8.18.1.3 RelationValidationFailure [3/3] . . . 90
8.18.2 Function Documentation . . . 91
8.18.2.1 validate_schema() [1/2] . . . 91
8.18.2.2 validate_schema() [2/2] . . . 91
8.19 serif::discretization::forms Namespace Reference . . . 91
8.19.1 Typedef Documentation . . . 94
8.19.1.1 MatrixValue . . . 94
8.19.1.2 OperandValue . . . 94
8.19.1.3 PartialDerivative . . . 94
8.19.1.4 VectorValue . . . 94
8.19.2 Function Documentation . . . 95
8.19.2.1 component() . . . 95
8.19.2.2 derivative_component() . . . 95
8.19.3 Variable Documentation . . . 96
8.19.3.1 component_count . . . 96
8.19.3.2 form_descriptor . . . 96
8.20 serif::discretization::functional_space Namespace Reference . . . 96
8.20.1 Enumeration Type Documentation . . . 98
8.20.1.1 Continuity . . . 98
8.20.1.2 Family . . . 98
8.20.1.3 Mapping . . . 98
8.20.1.4 Trace . . . 99
8.20.2 Variable Documentation . . . 99

8.20.2.1 Continuity2StringMap . . . 99
8.20.2.2 Family2StringMap . . . 99
8.20.2.3 H1 . . . 100
8.20.2.4 Hcurl . . . 100
8.20.2.5 Hdiv . . . 101
8.20.2.6 L2 . . . 101
8.20.2.7 Mapping2StringMap . . . 101
8.20.2.8 String2ContinuityMap . . . 102
8.20.2.9 String2FamilyMap . . . 102
8.20.2.10 String2MappingMap . . . 102
8.20.2.11 String2TraceMap . . . 102
8.20.2.12 Trace2StringMap . . . 102
8.21 serif::discretization::quadrature Namespace Reference . . . 102
8.21.1 Enumeration Type Documentation . . . 103
8.21.1.1 MappingKind . . . 103
8.21.1.2 Mode . . . 103
8.21.1.3 QuadratureRole . . . 104
8.21.2 Function Documentation . . . 104
8.21.2.1 make_query() . . . 104
8.22 serif::discretization::quadrature::backend Namespace Reference . . . 104
8.23 serif::discretization::quadrature::backend::mfem Namespace Reference . . . 105
8.23.1 Typedef Documentation . . . 105
8.23.1.1 FunctionalSpaceTraits . . . 105
8.23.2 Function Documentation . . . 106
8.23.2.1 resolve_operand_order() . . . 106
8.23.2.2 set_integration_rule() . . . 106
8.23.2.3 sum_dynamic_orders() . . . 106
8.23.3 Variable Documentation . . . 106
8.23.3.1 mfem_form_static_order . . . 106
8.24 serif::eos Namespace Reference . . . 106
8.24.1 Function Documentation . . . 107
8.24.1.1 evaluate() . . . 107
8.24.1.2 partial_derivative() . . . 107
8.25 serif::eos::models Namespace Reference . . . 107
8.25.1 Enumeration Type Documentation . . . 108
8.25.1.1 SERiFErrorCode . . . 108
8.25.2 Function Documentation . . . 109
8.25.2.1 validate_finite() . . . 109
8.25.2.2 validate_greater_than() . . . 109
8.25.2.3 validate_less_than() . . . 109

8.25.2.4 validate_nonnegativity() . . . 109
8.25.3 Variable Documentation . . . 109
8.25.3.1 is_equation_of_state_model_v . . . 109
8.25.3.2 supports_partial_derivative_v . . . 109
8.25.3.3 supports_relation_v . . . 110
8.26 serif::eos::relations Namespace Reference . . . 110
8.26.1 Typedef Documentation . . . 111
8.26.1.1 DensityFromSpecificEnthalpy . . . 111
8.26.1.2 PressureFromDensity . . . 111
8.26.1.3 PressureFromSpecificEnthalpy . . . 111
8.26.1.4 SpecificEnthalpyFromDensity . . . 111
8.26.1.5 SpecificEnthalpyFromPressure . . . 112
8.26.2 Function Documentation . . . 112
8.26.2.1 CountVariadicArguments() . . . 112
8.26.3 Variable Documentation . . . 112
8.26.3.1 is_thermodynamic_relation_v . . . 112
8.26.3.2 is_valid_relation_catalog_v . . . 112
8.26.3.3 relation_catalog_contains_relation_v . . . 112
8.26.3.4 relation_contains_input_v . . . 112
8.27 serif::eos::runtime Namespace Reference . . . 113
8.27.1 Typedef Documentation . . . 114
8.27.1.1 RuntimeCatalogQuantityTuple . . . 114
8.27.1.2 RuntimeEOSAdapter . . . 114
8.27.2 Function Documentation . . . 114
8.27.2.1 evaluateRuntimePartialDerivative() . . . 114
8.27.2.2 evaluateRuntimeRelation() . . . 114
8.27.2.3 runtimeEvaluationFailure() . . . 115
8.27.2.4 runtimeQuantityIdentifierIsUnambiguous() . . . 115
8.27.2.5 runtimeQuantityIdentifiersAreCompatible() . . . 115
8.27.2.6 runtimeQuantityIdentifiersAreUnambiguous() . . . 115
8.27.2.7 runtimeRelationDescriptors() . . . 115
8.27.2.8 runtimeRelationMatches() . . . 115
8.27.2.9 tryRuntimePartialDerivative() . . . 115
8.28 serif::equations Namespace Reference . . . 116
8.28.1 Typedef Documentation . . . 117
8.28.1.1 ContributionsOf . . . 117
8.28.1.2 SupportOf . . . 117
8.28.1.3 TestOperandOf . . . 117
8.29 serif::equations::gravity_contributions Namespace Reference . . . 117
8.29.1 Typedef Documentation . . . 118

8.29.1.1 Density . . . 118
8.29.1.2 Divergence . . . 118
8.29.1.3 Gradient . . . 118
8.29.1.4 Operand . . . 118
8.29.1.5 Potential . . . 118
8.29.1.6 Value . . . 118
8.30 serif::geometry Namespace Reference . . . 118
8.31 serif::geometry::deformation Namespace Reference . . . 119
8.31.1 Typedef Documentation . . . 122
8.31.1.1 DeformationStatus . . . 122
8.31.1.2 DeformationWorkspace . . . 122
8.31.2 Enumeration Type Documentation . . . 122
8.31.2.1 DeformationFailureReason . . . 122
8.31.3 Function Documentation . . . 122
8.31.3.1 apply_jacobian() [1/2] . . . 122
8.31.3.2 apply_jacobian() [2/2] . . . 123
8.31.3.3 apply_jacobian_transpose() [1/2] . . . 123
8.31.3.4 apply_jacobian_transpose() [2/2] . . . 123
8.31.3.5 apply_pullback_derivative() [1/2] . . . 123
8.31.3.6 apply_pullback_derivative() [2/2] . . . 124
8.31.3.7 build_displacement() . . . 124
8.31.3.8 build_surface_embedding() [1/2] . . . 124
8.31.3.9 build_surface_embedding() [2/2] . . . 124
8.31.3.10 compose_deformation() . . . 125
8.31.3.11 evaluate_displacement() . . . 125
8.31.3.12 make_deformation_computations() . . . 125
8.31.3.13 make_deformation_graph() . . . 125
8.31.3.14 make_deformation_workspace() . . . 125
8.31.3.15 make_radial_extension() . . . 126
8.31.3.16 validate_deformation_values() . . . 126
8.31.4 Variable Documentation . . . 126
8.31.4.1 is_linear_deformation . . . 126
8.32 serif::geometry::deformation::backend Namespace Reference . . . 126
8.33 serif::geometry::deformation::backend::mfem Namespace Reference . . . 126
8.33.1 Function Documentation . . . 128
8.33.1.1 build_radial_interpolation() . . . 128
8.33.1.2 deformation_discretization() . . . 129
8.33.1.3 estimate_geometry_step_limit() . . . 129
8.33.1.4 make_deformation_computations() . . . 129
8.33.1.5 make_deformation_graph() . . . 130

8.33.1.6 make_deformation_inputs() [1/2] . . . 130
8.33.1.7 make_deformation_inputs() [2/2] . . . 130
8.33.1.8 make_surface_metric_computation() . . . 130
8.33.1.9 make_volume_displacement_computations() [1/2] . . . 131
8.33.1.10 make_volume_displacement_computations() [2/2] . . . 131
8.33.1.11 sample_reference_positions() . . . 131
8.33.1.12 sample_reference_surface() . . . 131
8.33.1.13 select_surface_scalar_dofs() . . . 131
8.33.1.14 surface_parameter_mass_diagonal() . . . 132
8.33.1.15 validate_companion_numbering() . . . 132
8.33.1.16 validate_scalar_reference_space() . . . 132
8.34 serif::geometry::deformation::backend::mfem::detail Namespace Reference . . . 132
8.34.1 Function Documentation . . . 133
8.34.1.1 boundary_logical_radius() . . . 133
8.34.1.2 compile_radial_reference() . . . 133
8.34.1.3 locate_boundary_point() . . . 133
8.34.1.4 logical_radius() . . . 134
8.34.1.5 share_reference_result() . . . 134
8.35 serif::geometry::deformation::backend::mfem::volume_displacement_keys Namespace Reference 134
8.36 serif::geometry::deformation::detail Namespace Reference . . . 134
8.36.1 Typedef Documentation . . . 135
8.36.1.1 DeterminantPolynomial . . . 135
8.36.1.2 JacobianEntries . . . 135
8.36.2 Function Documentation . . . 135
8.36.2.1 check_independent_harmonics() . . . 135
8.36.2.2 determinant_polynomial() . . . 135
8.36.2.3 evaluate_polynomial() . . . 136
8.36.2.4 first_boundary_parameter() . . . 136
8.36.2.5 make_harmonic_modes() . . . 136
8.36.2.6 sample_harmonic_basis() . . . 136
8.36.2.7 validate_disjoint_output() . . . 136
8.37 serif::geometry::deformation::keys Namespace Reference . . . 137
8.38 serif::geometry::deformation::spaces Namespace Reference . . . 137
8.39 serif::geometry::surface Namespace Reference . . . 137
8.39.1 Typedef Documentation . . . 139
8.39.1.1 Isobaric . . . 139
8.39.1.2 pressure_relation_t . . . 139
8.39.1.3 SurfaceLinearization . . . 139
8.39.1.4 SurfacePoint . . . 139
8.39.2 Enumeration Type Documentation . . . 139

8.39.2.1 SurfaceFailureReason . . . 139
8.39.3 Function Documentation . . . 139
8.39.3.1 boundary_values() . . . 139
8.39.3.2 evaluate_pressure_surface() . . . 140
8.39.3.3 make_pressure_surface_graph() [1/2] . . . 140
8.39.3.4 make_pressure_surface_graph() [2/2] . . . 141
8.39.3.5 pressure_condition() . . . 141
8.40 serif::geometry::surface::detail Namespace Reference . . . 141
8.40.1 Typedef Documentation . . . 143
8.40.1.1 BoundTerms . . . 143
8.40.2 Function Documentation . . . 144
8.40.2.1 accumulate_partial() . . . 144
8.40.2.2 checked_linearization() . . . 144
8.40.2.3 evaluate_relation() . . . 144
8.40.2.4 evaluate_response_samples() . . . 144
8.40.2.5 evaluate_surface_response() . . . 145
8.40.2.6 finite_state() . . . 145
8.40.2.7 linearize_carrier_samples() . . . 145
8.40.2.8 linearize_pressure_samples() . . . 145
8.40.2.9 linearize_surface() . . . 146
8.40.2.10 make_surface_eos_computation() . . . 146
8.40.2.11 make_surface_linearization_computation() . . . 146
8.40.2.12 matches_pressure_relation() . . . 146
8.40.2.13 relation_input() . . . 146
8.40.2.14 state_value() . . . 147
8.40.2.15 supports_partials() . . . 147
8.40.2.16 surface_point() . . . 147
8.40.2.17 surface_sample_count() . . . 147
8.40.3 Variable Documentation . . . 147
8.40.3.1 is_pressure_surface_formulation . . . 147
8.40.3.2 is_pressure_surface_formulation< DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation > > . . . 147
8.40.3.3 is_pressure_surface_formulation< PressureSurfaceFormulation< EOS, Bindings, Carrier > > . . . 148
8.41 serif::geometry::surface::keys Namespace Reference . . . 148
8.42 serif::mapping Namespace Reference . . . 148
8.42.1 Enumeration Type Documentation . . . 150
8.42.1.1 FaceElementSide . . . 150
8.42.1.2 MappingStatus . . . 150
8.42.2 Function Documentation . . . 150

8.42.2.1 compute_hdiv_mass_tensor() . . . 150
8.42.2.2 compute_hdiv_mass_tensor_variation() . . . 150
8.42.2.3 compute_scalar_diffusion_tensor() . . . 151
8.42.2.4 element_displacement_data_from_element_vdofs() . . . 151
8.42.2.5 make_MappingStatus_SERiFErrorCode_map() . . . 151
8.42.2.6 make_SERIFErrorCode_MappingStatus_map() . . . 151
8.42.2.7 map_hcurl_curl_to_physical() . . . 151
8.42.2.8 map_hcurl_field_to_physical() . . . 151
8.42.2.9 map_hdiv_divergence_to_physical() . . . 151
8.42.2.10 map_hdiv_flux_to_physical() . . . 152
8.42.2.11 map_physical_curl_to_hcurl_reference() . . . 152
8.42.2.12 map_physical_field_to_hcurl_reference() . . . 152
8.42.2.13 map_physical_flux_to_hdiv_reference() . . . 152
8.42.2.14 map_physical_gradient_to_reference() . . . 152
8.42.2.15 map_physical_vector_gradient_to_reference() . . . 152
8.42.2.16 map_reference_gradient_to_physical() . . . 153
8.42.2.17 map_reference_vector_gradient_to_physical() . . . 153
8.42.3 Variable Documentation . . . 153
8.42.3.1 MappingStatus_enum_string_map . . . 153
8.42.3.2 MappingStatus_SERiFErrorCode_map . . . 153
8.42.3.3 MappingStatus_string_enum_map . . . 153
8.42.3.4 SERiFErrorCode_MappingStatus_map . . . 153
8.43 serif::mapping::compactification Namespace Reference . . . 153
8.44 serif::meta Namespace Reference . . . 154
8.44.1 Function Documentation . . . 154
8.44.1.1 enum_to_enum_string_map() . . . 154
8.44.1.2 enum_to_string() . . . 154
8.44.1.3 enum_to_string_enum_map() . . . 154
8.44.1.4 template_name() . . . 154
8.44.1.5 type_name() . . . 154
8.45 serif::operators Namespace Reference . . . 155
8.45.1 Typedef Documentation . . . 156
8.45.1.1 CentralDensity . . . 156
8.45.1.2 Closure . . . 156
8.45.1.3 Constant . . . 157
8.45.1.4 Density . . . 157
8.45.1.5 DensityValue . . . 157
8.45.1.6 Displacement . . . 157
8.45.1.7 DisplacementDivergence . . . 157
8.45.1.8 DisplacementValue . . . 157

8.45.1.9 Enthalpy . . . 157
8.45.1.10 EnthalpyGradient . . . 157
8.45.1.11 EnthalpyValue . . . 157
8.45.1.12 GravityDivergence . . . 157
8.45.1.13 GravityValue . . . 158
8.45.1.14 Mass . . . 158
8.45.1.15 Mechanical . . . 158
8.45.1.16 MechanicalDifference . . . 158
8.45.1.17 MixedSamples . . . 158
8.45.1.18 Momentum . . . 158
8.45.1.19 Omega . . . 158
8.45.1.20 Potential . . . 158
8.45.1.21 Pressure . . . 158
8.45.1.22 ScaledMechanical . . . 158
8.45.1.23 ScaledPoisson . . . 159
8.45.1.24 Schema . . . 159
8.45.1.25 SurfacePressure . . . 159
8.45.2 Function Documentation . . . 159
8.45.2.1 bind_contributions() . . . 159
8.45.2.2 for_each_contribution() . . . 159
8.46 serif::operators::backend Namespace Reference . . . 159
8.47 serif::operators::backend::mfem Namespace Reference . . . 159
8.47.1 Typedef Documentation . . . 169
8.47.1.1 Centering . . . 169
8.47.1.2 Closure . . . 170
8.47.1.3 Density . . . 170
8.47.1.4 DensityBinding . . . 170
8.47.1.5 DisplacementBinding . . . 170
8.47.1.6 ElementBatch . . . 170
8.47.1.7 EnthalpyBinding . . . 170
8.47.1.8 NodalCache . . . 170
8.47.1.9 NodalCoefficients . . . 170
8.47.1.10 NodalMesh . . . 171
8.47.1.11 PreparationMode . . . 171
8.47.1.12 SamplesFor . . . 171
8.47.1.13 Schema . . . 171
8.47.1.14 Vector . . . 171
8.47.1.15 VolumeCache . . . 171
8.47.1.16 VolumeCoefficients . . . 171
8.47.1.17 VolumeEquationOf . . . 172

8.47.1.18 VolumeInputs . . . 172
8.47.1.19 VolumeMesh . . . 172
8.47.1.20 VolumeOperandResult . . . 172
8.47.2 Enumeration Type Documentation . . . 172
8.47.2.1 ContractionMode [1/2] . . . 172
8.47.2.2 ContractionMode [2/2] . . . 172
8.47.2.3 PreparationRejectionReason . . . 172
8.47.2.4 ReferenceOperation . . . 173
8.47.3 Function Documentation . . . 173
8.47.3.1 accumulate_element_result() . . . 173
8.47.3.2 add_jacobian_transpose_rows() . . . 173
8.47.3.3 add_point_border_residual() . . . 173
8.47.3.4 apply_jacobian() [1/2] . . . 174
8.47.3.5 apply_jacobian() [2/2] . . . 174
8.47.3.6 apply_jacobian_rows() [1/2] . . . 174
8.47.3.7 apply_jacobian_rows() [2/2] . . . 174
8.47.3.8 apply_jacobian_transpose() . . . 175
8.47.3.9 apply_point_border_jacobian() . . . 175
8.47.3.10 apply_point_border_jacobian_transpose() . . . 175
8.47.3.11 apply_residual_rows() . . . 175
8.47.3.12 apply_volume_direction() . . . 176
8.47.3.13 bind() [1/4] . . . 176
8.47.3.14 bind() [2/4] . . . 176
8.47.3.15 bind() [3/4] . . . 176
8.47.3.16 bind() [4/4] . . . 176
8.47.3.17 collect_volume_results() . . . 177
8.47.3.18 connect_volume_contributions() . . . 177
8.47.3.19 element_equation_point() . . . 177
8.47.3.20 element_jacobian_action() . . . 177
8.47.3.21 element_jacobian_transpose() . . . 178
8.47.3.22 element_residual() . . . 178
8.47.3.23 element_values() . . . 178
8.47.3.24 equation_rule() . . . 178
8.47.3.25 evaluate_contribution_equation() . . . 179
8.47.3.26 evaluate_element_equation() . . . 179
8.47.3.27 evaluate_nodal_values() . . . 179
8.47.3.28 for_each_volume_element() . . . 179
8.47.3.29 geometry_integrand_variation() . . . 180
8.47.3.30 integration_order() . . . 180
8.47.3.31 make_boundary_discretization() . . . 180

8.47.3.32 make_contribution_computations() . . . 180
8.47.3.33 make_nodal_dependency_context() . . . 180
8.47.3.34 make_nodal_discretization() . . . 181
8.47.3.35 make_nodal_equation_cache() . . . 181
8.47.3.36 make_nodal_equation_computation() . . . 181
8.47.3.37 make_nodal_equation_point() . . . 181
8.47.3.38 make_operand_computations() . . . 181
8.47.3.39 make_point_border_cache() . . . 182
8.47.3.40 make_point_border_dependency_context() . . . 182
8.47.3.41 make_point_border_discretization() . . . 182
8.47.3.42 make_point_discretization() . . . 182
8.47.3.43 make_term_coefficients() . . . 182
8.47.3.44 make_volume_equation_cache() . . . 183
8.47.3.45 make_volume_equation_computations() . . . 183
8.47.3.46 make_volume_inputs() [1/6] . . . 183
8.47.3.47 make_volume_inputs() [2/6] . . . 183
8.47.3.48 make_volume_inputs() [3/6] . . . 183
8.47.3.49 make_volume_inputs() [4/6] . . . 184
8.47.3.50 make_volume_inputs() [5/6] . . . 184
8.47.3.51 make_volume_inputs() [6/6] . . . 184
8.47.3.52 map_contribution_geometry() . . . 184
8.47.3.53 map_contribution_operand() . . . 185
8.47.3.54 map_element_geometry() . . . 185
8.47.3.55 map_element_operand() . . . 185
8.47.3.56 mask_residual_dual() . . . 185
8.47.3.57 nodal_work_report() . . . 185
8.47.3.58 prepare_nodal() . . . 186
8.47.3.59 prepare_point_border() . . . 186
8.47.3.60 prepare_volume() . . . 186
8.47.3.61 report_contribution_preparation() . . . 186
8.47.3.62 sample_contribution_basis() . . . 187
8.47.3.63 sample_contribution_operand() . . . 187
8.47.3.64 sample_element_basis() . . . 187
8.47.3.65 select_nodal_values() . . . 187
8.47.3.66 store_element_linearization() . . . 188
8.47.3.67 store_nodal_linearization() . . . 188
8.47.3.68 tensor_element_count() . . . 188
8.47.3.69 try_prepare_nodal() . . . 188
8.47.3.70 try_prepare_point_border() . . . 189
8.47.3.71 try_prepare_validated() . . . 189

8.47.3.72 try_prepare_volume() . . . 189
8.47.3.73 validate_nodal_discretization() . . . 189
8.47.3.74 validate_nodal_output() . . . 189
8.47.3.75 validate_nodal_selection() . . . 190
8.47.3.76 validate_point_border_outputs() . . . 190
8.47.3.77 validate_volume_data() . . . 190
8.47.3.78 validate_volume_discretization() . . . 190
8.47.3.79 volume_height() . . . 190
8.47.3.80 volume_residual() . . . 191
8.47.3.81 volume_width() . . . 191
8.47.3.82 volume_work_report() . . . 191
8.47.3.83 VolumeDiscretization() . . . 191
8.47.4 Variable Documentation . . . 191
8.47.4.1 index . . . 191
8.47.4.2 values . . . 192
8.48 serif::operators::backend::mfem::nodal_keys Namespace Reference . . . 192
8.49 serif::operators::backend::mfem::volume_keys Namespace Reference . . . 192
8.50 serif::operators::detail Namespace Reference . . . 192
8.50.1 Function Documentation . . . 193
8.50.1.1 bind_contributions() . . . 193
8.50.1.2 visit_contributions() . . . 193
8.51 serif::utils Namespace Reference . . . 193
8.52 serif::utils::checks Namespace Reference . . . 193
8.52.1 Function Documentation . . . 194
8.52.1.1 is_finite() [1/3] . . . 194
8.52.1.2 is_finite() [2/3] . . . 194
8.52.1.3 is_finite() [3/3] . . . 194
8.53 serif::utils::dependencies Namespace Reference . . . 194
8.53.1 Typedef Documentation . . . 198
8.53.1.1 bound_type_t . . . 198
8.53.1.2 CombinedKeyList . . . 198
8.53.1.3 InputList . . . 198
8.53.1.4 RequestedResults . . . 198
8.53.1.5 ResolvedTargets . . . 198
8.53.2 Enumeration Type Documentation . . . 198
8.53.2.1 PreparationMode . . . 198
8.53.2.2 PreparationStatus . . . 198
8.53.3 Function Documentation . . . 199
8.53.3.1 borrow() [1/2] . . . 199
8.53.3.2 borrow() [2/2] . . . 199

8.53.3.3 make_computation() . . . 199
8.53.3.4 make_computation_cache() [1/2] . . . 199
8.53.3.5 make_computation_cache() [2/2] . . . 200
8.53.3.6 make_computation_graph() . . . 200
8.53.3.7 make_computation_store() . . . 200
8.53.3.8 make_dependency_context() . . . 200
8.53.3.9 make_incremental_computation() . . . 200
8.53.3.10 own() . . . 201
8.53.3.11 own_computation() . . . 201
8.53.3.12 source_identity() . . . 201
8.53.3.13 track_by() . . . 201
8.53.4 Variable Documentation . . . 201
8.53.4.1 has_binding_v . . . 201
8.53.4.2 is_bindings_v . . . 201
8.53.4.3 is_bindings_v< Bindings< BindingTs... > > . . . 202
8.53.4.4 is_computation_graph_v . . . 202
8.53.4.5 is_computation_graph_v< ComputationGraph< Computations... > > . . . 202
8.53.4.6 is_computation_v . . . 202
8.53.4.7 is_computation_v< Computation< KeyT, InputsT, FunctionT > > . . . 202
8.53.4.8 is_computation_v< IncrementalComputation< KeyT, ValueT, InputsT, FunctionT > > . . . 202
8.53.4.9 is_dependency_context_v . . . 202
8.53.4.10 is_dependency_context_v< DependencyContext< Resources... > > . . . 202
8.53.4.11 is_dependency_context_v< DependencyContextView< ParentT, KeysT > > 203
8.53.4.12 is_key_list_v . . . 203
8.53.4.13 is_key_list_v< KeyList< KeyTs... > > . . . 203
8.53.4.14 is_owned_computation_v . . . 203
8.53.4.15 is_owned_computation_v< OwnedComputation< KeyT, ComputationT > > 203
8.53.4.16 is_resource_v . . . 203
8.53.4.17 is_resource_v< Resource< KeyT, StorageT, PolicyT > > . . . 203
8.53.4.18 preparation_mode_name_map . . . 203
8.53.4.19 preparation_status_name_map . . . 204
8.54 serif::utils::dependencies::detail Namespace Reference . . . 204
8.54.1 Typedef Documentation . . . 205
8.54.1.1 UsedResources . . . 205
8.54.2 Function Documentation . . . 206
8.54.2.1 compatible_errors() . . . 206
8.54.2.2 independent_results() . . . 206
8.54.2.3 type_index() . . . 206
8.54.3 Variable Documentation . . . 206

8.54.3.1 compatible_error_v . . . 206
8.54.3.2 compatible_error_v< Source, std::variant< Alternatives... > > . . . 206
8.54.3.3 contains_key_v . . . 207
8.54.3.4 contains_key_v< KeyT, KeyList< Keys... > > . . . 207
8.54.3.5 key_list_subset_v . . . 207
8.54.3.6 key_list_subset_v< KeyList< Keys... >, AvailableT > . . . 207
8.55 serif::utils::dependencies::keys Namespace Reference . . . 207
8.56 serif::utils::errors Namespace Reference . . . 207
8.56.1 Enumeration Type Documentation . . . 208
8.56.1.1 SERiFErrorCode . . . 208
8.56.2 Function Documentation . . . 210
8.56.2.1 serif_assert() [1/2] . . . 210
8.56.2.2 serif_assert() [2/2] . . . 210
8.56.2.3 serif_ensure() [1/2] . . . 210
8.56.2.4 serif_ensure() [2/2] . . . 210
8.56.2.5 serif_error() . . . 210
8.56.3 Variable Documentation . . . 211
8.56.3.1 SERiFErrorCodeNameToTypeMap . . . 211
8.56.3.2 SERiFErrorCodeTypeToNameMap . . . 211
8.57 serif::utils::misc Namespace Reference . . . 211
8.57.1 Function Documentation . . . 211
8.57.1.1 validate_finite() . . . 211
8.57.1.2 validate_greater_than() . . . 212
8.57.1.3 validate_less_than() . . . 212
8.57.1.4 validate_nonnegativity() . . . 212
8.58 serif::utils::misc::concepts Namespace Reference . . . 212
8.58.1 Typedef Documentation . . . 213
8.58.1.1 ConcatT . . . 213
8.58.1.2 EnumerateTypesT . . . 213
8.58.1.3 FilterT . . . 213
8.58.1.4 PrependT . . . 213
8.59 serif::utils::misc::format Namespace Reference . . . 213
8.59.1 Function Documentation . . . 214
8.59.1.1 join() . . . 214
8.60 serif::utils::misc::std_helper Namespace Reference . . . 214
8.60.1 Typedef Documentation . . . 214
8.60.1.1 NeitherConstNorVolatile . . . 214
8.60.1.2 NeitherConstNorVolatileReference . . . 214
8.60.2 Function Documentation . . . 215
8.60.2.1 CountVariadicArguments() . . . 215

8.60.3 Variable Documentation . . . 215
8.60.3.1 AIsBaseClassOfB . . . 215
8.60.3.2 SameType . . . 215
8.61 serif::utils::misc::terminal Namespace Reference . . . 215
8.61.1 Enumeration Type Documentation . . . 216
8.61.1.1 ANSIColors . . . 216
8.61.2 Function Documentation . . . 217
8.61.2.1 resolve_ansi_color() . . . 217
8.61.2.2 supports_ansi() . . . 217
8.61.3 Variable Documentation . . . 217
8.61.3.1 a_black . . . 217
8.61.3.2 a_blue . . . 217
8.61.3.3 a_bright_black . . . 217
8.61.3.4 a_bright_green . . . 218
8.61.3.5 a_bright_red . . . 218
8.61.3.6 a_cyan . . . 218
8.61.3.7 a_green . . . 218
8.61.3.8 a_magenta . . . 218
8.61.3.9 a_red . . . 218
8.61.3.10 a_reset . . . 218
8.61.3.11 a_white . . . 218
8.61.3.12 a_yellow . . . 218
8.62 serif::utils::misc::traits Namespace Reference . . . 219
8.62.1 Variable Documentation . . . 219
8.62.1.1 all_unique_types_v . . . 219
8.62.1.2 all_unique_types_v< First, Rest... > . . . 219
8.63 serif::utils::types Namespace Reference . . . 219
8.64 serif::utils::types::backend Namespace Reference . . . 219
8.65 serif::utils::types::backend::mfem Namespace Reference . . . 220
8.66 serif::utils::user_options Namespace Reference . . . 220
8.67 serif::utils::warnings Namespace Reference . . . 220
8.67.1 Function Documentation . . . 220
8.67.1.1 serif_warning() . . . 220

9 Concept Documentation 221
9.1 @1200271712111733323463461531470020022660652232 66::HasBoundType Concept Reference . . . 221
9.1.1 Concept definition . . . 221
9.2 @1200271712111733323463461531470020022660652232 66::HasBoundTypeMember Concept Reference . . . 221
9.2.1 Concept definition . . . 221
9.3 @2000743173031670560630642560700462030540640522 14::CanInspectChange Concept Reference 221

9.3.1 Concept definition . . . 221
9.4 @20007431730316705606306425607004620305406405221 4::CanMakeCache Concept Reference . 222
9.4.1 Concept definition . . . 222
9.5 @20007431730316705606306425607004620305406405221 4::CanMakeComputation Concept Reference . . . 222
9.5.1 Concept definition . . . 222
9.6 @20007431730316705606306425607004620305406405221 4::CanMakeFallibleCache Concept Reference . . . 222
9.6.1 Concept definition . . . 222
9.7 @20007431730316705606306425607004620305406405221 4::CanRequest Concept Reference . . . 222
9.7.1 Concept definition . . . 222
9.8 @23037207512511711214621602205025733117713710732 7::CanCompose Concept Reference . . . 222
9.8.1 Concept definition . . . 223
9.8.2 Detailed Description . . . 223
9.9 @24700627707721121615335034114235733716100102614 3::HasSurfaceRelation Concept Reference 223
9.9.1 Concept definition . . . 223
9.10 @24700627707721121615335034114235733716100102614 3::SelectsCarrier Concept Reference . . 223
9.10.1 Concept definition . . . 223
9.11 @25216614303526431402121424602605504112016332204 1::CanResolve Concept Reference . . . 223
9.11.1 Concept definition . . . 223
9.12 @32317232736410223521033333705706501515235433134 3::CanBorrow Concept Reference . . . 223
9.12.1 Concept definition . . . 223
9.13 @32317232736410223521033333705706501515235433134 3::CanMakeContext Concept Reference 224
9.13.1 Concept definition . . . 224
9.14 @32317232736410223521033333705706501515235433134 3::CanRead Concept Reference 224
9.14.1 Concept definition . . . 224
9.15 @32317232736410223521033333705706501515235433134 3::CanSelect Concept Reference 224
9.15.1 Concept definition . . . 224
9.16 @34714524103410214107200435717127436023034606435 6::CanMakeStore Concept Reference . 224
9.16.1 Concept definition . . . 224
9.17 @34714524103410214107200435717127436023034606435 6::CanOwn Concept Reference 224
9.17.1 Concept definition . . . 224
9.18 @34714524103410214107200435717127436023034606435 6::CanRead Concept Reference 225
9.18.1 Concept definition . . . 225
9.19 @34714524103410214107200435717127436023034606435 6::CanReadEntry Concept Reference . 225
9.19.1 Concept definition . . . 225
9.20 serif::dimensions::PhysicalQuantityType Concept Reference . . . 225
9.20.1 Concept definition . . . 225
9.20.2 Detailed Description . . . 225
9.21 serif::dimensions::QuantityValueType Concept Reference . . . 226

9.21.1 Concept definition . . . 226
9.21.2 Detailed Description . . . 226
9.22 serif::dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity Concept Reference . . . 226
9.22.1 Concept definition . . . 226
9.22.2 Detailed Description . . . 226
9.23 serif::dimensions::ThermodynamicQuantityType Concept Reference . . . 226
9.23.1 Concept definition . . . 227
9.23.2 Detailed Description . . . 227
9.24 serif::dimensions::ThermodynamicQuantityValueType Concept Reference . . . 227
9.24.1 Concept definition . . . 227
9.25 serif::discretization::blocks::Field Concept Reference . . . 227
9.25.1 Concept definition . . . 227
9.25.2 Detailed Description . . . 227
9.26 serif::discretization::blocks::GlobalScalarTerm Concept Reference . . . 228
9.26.1 Concept definition . . . 228
9.26.2 Detailed Description . . . 228
9.27 serif::discretization::blocks::HasFunctionalSpace Concept Reference . . . 228
9.27.1 Concept definition . . . 228
9.27.2 Detailed Description . . . 228
9.28 serif::discretization::blocks::HasQuantity Concept Reference . . . 228
9.28.1 Concept definition . . . 229
9.28.2 Detailed Description . . . 229
9.29 serif::discretization::blocks::HasResidualBlock Concept Reference . . . 229
9.29.1 Concept definition . . . 229
9.29.2 Detailed Description . . . 229
9.30 serif::discretization::blocks::HasSupport Concept Reference . . . 229
9.30.1 Concept definition . . . 229
9.30.2 Detailed Description . . . 230
9.31 serif::discretization::blocks::HasValueBlock Concept Reference . . . 230
9.31.1 Concept definition . . . 230
9.31.2 Detailed Description . . . 230
9.32 serif::discretization::blocks::PhysicalSpatialTerm Concept Reference . . . 230
9.32.1 Concept definition . . . 230
9.32.2 Detailed Description . . . 230
9.33 serif::discretization::blocks::SpatialTerm Concept Reference . . . 231
9.33.1 Concept definition . . . 231
9.33.2 Detailed Description . . . 231
9.34 serif::discretization::blocks::Term Concept Reference . . . 231
9.34.1 Concept definition . . . 231
9.34.2 Detailed Description . . . 231

9.35 serif::discretization::domain::BoundaryIsIdentifiedBy Concept Reference . . . 231
9.35.1 Concept definition . . . 232
9.36 serif::discretization::domain::DomainIsIdentifiedBy Concept Reference . . . 232
9.36.1 Concept definition . . . 232
9.37 serif::discretization::domain::ids::IsBoundaryID Concept Reference . . . 232
9.37.1 Concept definition . . . 232
9.37.2 Detailed Description . . . 232
9.38 serif::discretization::domain::ids::IsDomainID Concept Reference . . . 232
9.38.1 Concept definition . . . 232
9.38.2 Detailed Description . . . 233
9.39 serif::discretization::domain::ids::IsDomainOrBoundaryID Concept Reference . . . 233
9.39.1 Concept definition . . . 233
9.39.2 Detailed Description . . . 233
9.40 serif::discretization::domain::ids::lists::IsBoundaryIDList Concept Reference . . . 233
9.40.1 Concept definition . . . 233
9.40.2 Detailed Description . . . 233
9.41 serif::discretization::domain::ids::lists::IsDomainIDList Concept Reference . . . 234
9.41.1 Concept definition . . . 234
9.41.2 Detailed Description . . . 234
9.42 serif::discretization::domain::ids::validation::AreDistinctBoundaryIDs Concept Reference . . . 234
9.42.1 Concept definition . . . 234
9.42.2 Detailed Description . . . 234
9.43 serif::discretization::domain::ids::validation::AreDistinctDomainIDs Concept Reference . . . 235
9.43.1 Concept definition . . . 235
9.43.2 Detailed Description . . . 235
9.44 serif::discretization::domain::ids::validation::HaveUniqueBoundaries Concept Reference . . . 235
9.44.1 Concept definition . . . 235
9.44.2 Detailed Description . . . 235
9.45 serif::discretization::domain::ids::validation::HaveUniqueBoundaryIDs Concept Reference . . . 236
9.45.1 Concept definition . . . 236
9.45.2 Detailed Description . . . 236
9.46 serif::discretization::domain::ids::validation::HaveUniqueDomainIDs Concept Reference . . . 236
9.46.1 Concept definition . . . 236
9.46.2 Detailed Description . . . 236
9.47 serif::discretization::domain::ids::validation::HaveUniqueDomains Concept Reference . . . 237
9.47.1 Concept definition . . . 237
9.47.2 Detailed Description . . . 237
9.48 serif::discretization::domain::IsBoundary Concept Reference . . . 237
9.48.1 Concept definition . . . 237
9.49 serif::discretization::domain::IsDomain Concept Reference . . . 237

9.49.1 Concept definition . . . 238
9.50 serif::discretization::domain::IsDomainOrSet Concept Reference . . . 238
9.50.1 Concept definition . . . 238
9.51 serif::discretization::domain::IsDomainSet Concept Reference . . . 238
9.51.1 Concept definition . . . 238
9.52 serif::discretization::domain::relation::IsRelation Concept Reference . . . 238
9.52.1 Concept definition . . . 238
9.53 serif::discretization::domain::relation::lists::IsRelationList Concept Reference . . . 238
9.53.1 Concept definition . . . 238
9.54 serif::discretization::domain::relation::validation::HaveValidRelationEntities Concept Reference 239
9.54.1 Concept definition . . . 239
9.55 serif::discretization::domain::schema::IsSchema Concept Reference . . . 239
9.55.1 Concept definition . . . 239
9.56 serif::discretization::forms::Form Concept Reference . . . 239
9.56.1 Concept definition . . . 239
9.56.2 Detailed Description . . . 239
9.57 serif::discretization::forms::IsOperand Concept Reference . . . 240
9.57.1 Concept definition . . . 240
9.57.2 Detailed Description . . . 240
9.58 serif::discretization::forms::IsOperation Concept Reference . . . 240
9.58.1 Concept definition . . . 240
9.58.2 Detailed Description . . . 240
9.59 serif::discretization::functional_space::IsSpaceDescriptor Concept Reference . . . 241
9.59.1 Concept definition . . . 241
9.59.2 Detailed Description . . . 241
9.60 serif::discretization::quadrature::backend::mfem::ConfigurableIntegrator Concept Reference . . . 241
9.60.1 Concept definition . . . 241
9.61 serif::discretization::quadrature::backend::mfem::HasPointerSetIntegrationRule Concept Reference . . . 241
9.61.1 Concept definition . . . 241
9.62 serif::discretization::quadrature::backend::mfem::HasReferenceSetIntegrationRule Concept Reference . . . 242
9.62.1 Concept definition . . . 242
9.63 serif::discretization::quadrature::backend::mfem::HasSetIntRule Concept Reference . . . 242
9.63.1 Concept definition . . . 242
9.64 serif::eos::models::BarotropicClosureEOS Concept Reference . . . 242
9.64.1 Concept definition . . . 242
9.65 serif::eos::models::EOSModel Concept Reference . . . 242
9.65.1 Concept definition . . . 242
9.66 serif::eos::models::EOSSupportingPartialDerivative Concept Reference . . . 243

9.66.1 Concept definition . . . 243
9.67 serif::eos::models::EOSSupportingRelation Concept Reference . . . 243
9.67.1 Concept definition . . . 243
9.68 serif::eos::models::PressureForceEOS Concept Reference . . . 243
9.68.1 Concept definition . . . 243
9.69 serif::eos::models::StructureSeedEOS Concept Reference . . . 243
9.69.1 Concept definition . . . 243
9.70 serif::eos::relations::CatalogContainingRelation Concept Reference . . . 243
9.70.1 Concept definition . . . 244
9.71 serif::eos::relations::RelationContainingInput Concept Reference . . . 244
9.71.1 Concept definition . . . 244
9.72 serif::eos::relations::ValidRelationCatalog Concept Reference . . . 244
9.72.1 Concept definition . . . 244
9.73 serif::eos::relations::ValidThermodynamicRelation Concept Reference . . . 244
9.73.1 Concept definition . . . 244
9.74 serif::eos::runtime::RuntimeEOSModel Concept Reference . . . 244
9.74.1 Concept definition . . . 244
9.75 serif::equations::Equation Concept Reference . . . 245
9.75.1 Concept definition . . . 245
9.75.2 Detailed Description . . . 245
9.76 serif::equations::PointEquation Concept Reference . . . 245
9.76.1 Concept definition . . . 245
9.77 serif::geometry::deformation::DeformationMap Concept Reference . . . 245
9.77.1 Concept definition . . . 245
9.78 serif::geometry::deformation::DifferentiableDeformationMap Concept Reference . . . 246
9.78.1 Concept definition . . . 246
9.78.2 Detailed Description . . . 246
9.79 serif::geometry::deformation::LinearDeformationMap Concept Reference . . . 246
9.79.1 Concept definition . . . 246
9.79.2 Detailed Description . . . 246
9.80 serif::geometry::surface::DirectPressureSurfaceCompilable Concept Reference . . . 247
9.80.1 Concept definition . . . 247
9.80.2 Detailed Description . . . 247
9.81 serif::geometry::surface::IsobaricSurfaceType Concept Reference . . . 247
9.81.1 Concept definition . . . 247
9.82 serif::geometry::surface::PressureSurfaceCompilable Concept Reference . . . 247
9.82.1 Concept definition . . . 247
9.82.2 Detailed Description . . . 247
9.83 serif::geometry::surface::PressureSurfaceFormulationType Concept Reference . . . 248
9.83.1 Concept definition . . . 248

9.84 serif::geometry::surface::SurfaceResidualRow Concept Reference . . . 248
9.84.1 Concept definition . . . 248
9.84.2 Detailed Description . . . 248
9.85 serif::geometry::surface::SurfaceScalarTerm Concept Reference . . . 248
9.85.1 Concept definition . . . 249
9.85.2 Detailed Description . . . 249
9.86 serif::geometry::surface::SurfaceStateBindings Concept Reference . . . 249
9.86.1 Concept definition . . . 249
9.86.2 Detailed Description . . . 249
9.87 serif::operators::backend::mfem::AffinePointEquation Concept Reference . . . 249
9.87.1 Concept definition . . . 249
9.87.2 Detailed Description . . . 250
9.88 serif::operators::backend::mfem::ClosureInputsAccept Concept Reference . . . 250
9.88.1 Concept definition . . . 250
9.89 serif::operators::backend::mfem::ClosureResult Concept Reference . . . 250
9.89.1 Concept definition . . . 250
9.90 serif::operators::backend::mfem::EvaluableOperand Concept Reference . . . 250
9.90.1 Concept definition . . . 250
9.90.2 Detailed Description . . . 251
9.91 serif::operators::CanSubtract Concept Reference . . . 251
9.91.1 Concept definition . . . 251
9.92 serif::operators::CanSum Concept Reference . . . 251
9.92.1 Concept definition . . . 251
9.93 serif::operators::NodalValueEquation Concept Reference . . . 251
9.93.1 Concept definition . . . 251
9.93.2 Detailed Description . . . 252
9.94 serif::operators::ScalarValueOperand Concept Reference . . . 252
9.94.1 Concept definition . . . 252
9.95 serif::operators::ScalarValueTerm Concept Reference . . . 252
9.95.1 Concept definition . . . 252
9.96 serif::operators::ScalarVolumeEquation Concept Reference . . . 252
9.96.1 Concept definition . . . 252
9.96.2 Detailed Description . . . 252
9.97 serif::operators::VolumeEquation Concept Reference . . . 253
9.97.1 Concept definition . . . 253
9.97.2 Detailed Description . . . 253
9.98 serif::operators::VolumeOperand Concept Reference . . . 253
9.98.1 Concept definition . . . 253
9.98.2 Detailed Description . . . 253
9.99 serif::utils::dependencies::BindingsWithBinding Concept Reference . . . 253

9.99.1 Concept definition . . . 254
9.100 serif::utils::dependencies::CanPrepare Concept Reference . . . 254
9.100.1 Concept definition . . . 254
9.101 serif::utils::dependencies::CompatibleComputations Concept Reference . . . 254
9.101.1 Concept definition . . . 254
9.102 serif::utils::dependencies::ComputationGraphFor Concept Reference . . . 254
9.102.1 Concept definition . . . 254
9.103 serif::utils::dependencies::IsBinding Concept Reference . . . 254
9.103.1 Concept definition . . . 255
9.104 serif::utils::dependencies::IsBindings Concept Reference . . . 255
9.104.1 Concept definition . . . 255
9.105 serif::utils::dependencies::IsComputation Concept Reference . . . 255
9.105.1 Concept definition . . . 255
9.106 serif::utils::dependencies::IsComputationGraph Concept Reference . . . 255
9.106.1 Concept definition . . . 255
9.107 serif::utils::dependencies::IsDependencyContext Concept Reference . . . 255
9.107.1 Concept definition . . . 255
9.108 serif::utils::dependencies::IsKeyList Concept Reference . . . 256
9.108.1 Concept definition . . . 256
9.109 serif::utils::dependencies::IsOwnedComputation Concept Reference . . . 256
9.109.1 Concept definition . . . 256
9.110 serif::utils::dependencies::IsResource Concept Reference . . . 256
9.110.1 Concept definition . . . 256
9.111 serif::utils::dependencies::SatisfiesKeyList Concept Reference . . . 256
9.111.1 Concept definition . . . 256
9.112 serif::utils::dependencies::SnapshotPolicyFor Concept Reference . . . 256
9.112.1 Concept definition . . . 257
9.113 serif::utils::misc::concepts::AllTypesAreUnique Concept Reference . . . 257
9.113.1 Concept definition . . . 257
9.114 serif::utils::misc::concepts::IsComparableToNumeric Concept Reference . . . 257
9.114.1 Concept definition . . . 257
9.115 serif::utils::misc::concepts::IsEnum Concept Reference . . . 257
9.115.1 Concept definition . . . 257
9.116 serif::utils::misc::concepts::IsException Concept Reference . . . 257
9.116.1 Concept definition . . . 258
9.117 serif::utils::misc::concepts::IsExceptionWithErrorCode Concept Reference . . . 258
9.117.1 Concept definition . . . 258
9.118 serif::utils::misc::concepts::IsNumeric Concept Reference . . . 258
9.118.1 Concept definition . . . 258
9.119 serif::utils::misc::concepts::IsOrderable Concept Reference . . . 258

9.119.1 Concept definition . . . 258
9.120 serif::utils::misc::concepts::SetOfOneOrTwo Concept Reference . . . 258
9.120.1 Concept definition . . . 258

10 Class Documentation 259
10.1 serif::dimensions::Acceleration Struct Reference . . . 259
10.1.1 Detailed Description . . . 260
10.1.2 Member Data Documentation . . . 260
10.1.2.1 identifier . . . 260
10.2 serif::discretization::blocks::rotation::angular_velocity Struct Reference . . . 261
10.2.1 Detailed Description . . . 262
10.2.2 Member Typedef Documentation . . . 262
10.2.2.1 Quantity . . . 262
10.2.2.2 Support . . . 262
10.2.3 Member Data Documentation . . . 262
10.2.3.1 name . . . 262
10.2.3.2 symbol . . . 263
10.3 serif::dimensions::AngularMomentum Struct Reference . . . 263
10.3.1 Detailed Description . . . 264
10.3.2 Member Data Documentation . . . 264
10.3.2.1 identifier . . . 264
10.4 serif::equations::AngularMomentum Class Reference . . . 265
10.4.1 Detailed Description . . . 266
10.4.2 Member Typedef Documentation . . . 266
10.4.2.1 AngularVelocity . . . 266
10.4.2.2 AngularVelocityValue . . . 266
10.4.2.3 Density . . . 266
10.4.2.4 DensityValue . . . 267
10.4.2.5 Operands . . . 267
10.4.3 Constructor & Destructor Documentation . . . 267
10.4.3.1 AngularMomentum() . . . 267
10.4.4 Member Function Documentation . . . 267
10.4.4.1 additional_order() [1/2] . . . 267
10.4.4.2 additional_order() [2/2] . . . 267
10.4.4.3 evaluate() . . . 267
10.4.4.4 residual_offset() . . . 267
10.4.4.5 validate_integral() . . . 268
10.4.4.6 validation_integrand() . . . 268
10.4.5 Member Data Documentation . . . 268
10.4.5.1 m_rotation . . . 268

10.4.5.2 m_target_momentum . . . 268
10.4.5.3 name . . . 268
10.5 serif::dimensions::AngularVelocity Struct Reference . . . 269
10.5.1 Detailed Description . . . 270
10.5.2 Member Data Documentation . . . 270
10.5.2.1 identifier . . . 270
10.6 serif::utils::dependencies::detail::AppendBinding< BindingsT, BindingT > Struct Template Reference . . . 270
10.7 serif::utils::dependencies::detail::AppendBinding< Bindings< BindingsTs... >, BindingT > Struct Template Reference . . . 270
10.7.1 Member Typedef Documentation . . . 270
10.7.1.1 type . . . 270
10.8 serif::utils::dependencies::detail::AppendKeys< ResultT, Keys > Struct Template Reference . . . 271
10.8.1 Member Typedef Documentation . . . 271
10.8.1.1 type . . . 271
10.9 serif::utils::dependencies::detail::AppendKeys< ResultT, FirstT, Rest... > Struct Template Reference . . . 271
10.9.1 Member Typedef Documentation . . . 272
10.9.1.1 type . . . 272
10.10 serif::utils::dependencies::detail::AppendUnique< ListT, T > Struct Template Reference . . . 272
10.11 serif::utils::dependencies::detail::AppendUnique< KeyList< Keys... >, T > Struct Template Reference . . . 272
10.11.1 Member Typedef Documentation . . . 273
10.11.1.1 type . . . 273
10.12 serif::dimensions::Area Struct Reference . . . 273
10.12.1 Detailed Description . . . 274
10.12.2 Member Data Documentation . . . 274
10.12.2.1 identifier . . . 274
10.13 serif::geometry::surface::detail::AreSurfaceBindings< Bindings > Struct Template Reference . . . 275
10.14 serif::geometry::surface::detail::AreSurfaceBindings< utils::dependencies::Bindings< Entries... > > Struct Template Reference . . . 276
10.15 serif::operators::detail::AsKeyList< ListT > Struct Template Reference . . . 276
10.16 serif::operators::detail::AsKeyList< utils::types::TypeList< T... > > Struct Template Reference . . . 277
10.16.1 Member Typedef Documentation . . . 277
10.16.1.1 type . . . 277
10.17 serif::operators::BadContributionCollection Struct Reference . . . 277
10.17.1 Member Typedef Documentation . . . 278
10.17.1.1 Test . . . 278
10.17.2 Member Function Documentation . . . 278
10.17.2.1 contributions() . . . 278
10.17.3 Member Data Documentation . . . 278

10.17.3.1 name . . . 278
10.18 serif::operators::BadContributionRow Struct Reference . . . 279
10.18.1 Member Typedef Documentation . . . 279
10.18.1.1 Test . . . 279
10.18.2 Member Function Documentation . . . 280
10.18.2.1 contributions() . . . 280
10.18.3 Member Data Documentation . . . 280
10.18.3.1 name . . . 280
10.19 serif::discretization::blocks::barotropic_constant Struct Reference . . . 280
10.19.1 Detailed Description . . . 281
10.19.2 Member Typedef Documentation . . . 281
10.19.2.1 Terms . . . 281
10.19.3 Member Data Documentation . . . 282
10.19.3.1 mass_normalization_term . . . 282
10.19.3.2 name . . . 282
10.20 serif::equations::BarotropicClosure< EOST > Class Template Reference . . . 282
10.20.1 Member Typedef Documentation . . . 283
10.20.1.1 Enthalpy . . . 283
10.20.1.2 Operands . . . 283
10.20.1.3 Test . . . 283
10.20.2 Constructor & Destructor Documentation . . . 284
10.20.2.1 BarotropicClosure() [1/2] . . . 284
10.20.2.2 BarotropicClosure() [2/2] . . . 284
10.20.3 Member Function Documentation . . . 284
10.20.3.1 additional_order() . . . 284
10.20.3.2 evaluate() . . . 284
10.20.4 Member Data Documentation . . . 284
10.20.4.1 m_enthalpy_power . . . 284
10.20.4.2 m_eos . . . 284
10.20.4.3 name . . . 285
10.21 serif::equations::BarotropicHydrostaticEquilibrium Class Reference . . . 285
10.21.1 Member Typedef Documentation . . . 286
10.21.1.1 AngularVelocity . . . 286
10.21.1.2 Constant . . . 286
10.21.1.3 Operands . . . 286
10.21.1.4 Potential . . . 286
10.21.1.5 Test . . . 286
10.21.2 Constructor & Destructor Documentation . . . 286
10.21.2.1 BarotropicHydrostaticEquilibrium() . . . 286
10.21.3 Member Function Documentation . . . 287

10.21.3.1 additional_order() . . . 287
10.21.3.2 evaluate() . . . 287
10.21.4 Member Data Documentation . . . 287
10.21.4.1 m_rotation . . . 287
10.21.4.2 name . . . 287
10.22 serif::operators::BarotropicHydrostaticEquilibrium Class Reference . . . 287
10.22.1 Member Typedef Documentation . . . 288
10.22.1.1 AngularVelocity . . . 288
10.22.1.2 Constant . . . 288
10.22.1.3 Operands . . . 288
10.22.1.4 Potential . . . 289
10.22.1.5 Test . . . 289
10.22.2 Constructor & Destructor Documentation . . . 289
10.22.2.1 BarotropicHydrostaticEquilibrium() . . . 289
10.22.3 Member Function Documentation . . . 289
10.22.3.1 additional_order() . . . 289
10.22.3.2 evaluate() . . . 289
10.22.4 Member Data Documentation . . . 289
10.22.4.1 m_rotation . . . 289
10.22.4.2 name . . . 290
10.23 serif::operators::backend::mfem::volume_keys::Basis< Contribution > Struct Template Reference . . . 290
10.24 serif::operators::backend::mfem::BasisBindings< ColumnCount > Struct Template Reference 290
10.24.1 Member Function Documentation . . . 291
10.24.1.1 operator==() . . . 291
10.24.2 Member Data Documentation . . . 291
10.24.2.1 columns . . . 291
10.24.2.2 compactification_space . . . 291
10.24.2.3 displacement_binding . . . 291
10.24.2.4 mapper . . . 291
10.24.2.5 nodes_sequence . . . 291
10.24.2.6 residual_binding . . . 292
10.25 serif::discretization::backend::mfem::BasisEvaluation Class Reference . . . 292
10.25.1 Detailed Description . . . 294
10.25.2 Member Function Documentation . . . 296
10.25.2.1 evaluate() . . . 296
10.25.2.2 height() . . . 296
10.25.2.3 integrate() . . . 296
10.25.2.4 map_point() . . . 296
10.25.2.5 reset_from() . . . 297

10.25.2.6 sample() . . . 297
10.25.2.7 set_constant() . . . 298
10.25.2.8 uses_tensor() . . . 298
10.25.2.9 width() . . . 298
10.25.3 Member Data Documentation . . . 298
10.25.3.1 m_components . . . 298
10.25.3.2 m_dense . . . 299
10.25.3.3 m_mapping . . . 299
10.25.3.4 m_points . . . 299
10.25.3.5 m_tensor . . . 299
10.25.3.6 m_width . . . 299
10.26 serif::operators::backend::mfem::BasisEvaluation Class Reference . . . 300
10.26.1 Detailed Description . . . 302
10.26.2 Member Function Documentation . . . 303
10.26.2.1 evaluate() . . . 303
10.26.2.2 height() . . . 303
10.26.2.3 integrate() . . . 303
10.26.2.4 map_point() . . . 304
10.26.2.5 reset_from() . . . 304
10.26.2.6 sample() . . . 305
10.26.2.7 set_constant() . . . 305
10.26.2.8 uses_tensor() . . . 305
10.26.2.9 width() . . . 305
10.26.3 Member Data Documentation . . . 306
10.26.3.1 m_components . . . 306
10.26.3.2 m_dense . . . 306
10.26.3.3 m_mapping . . . 306
10.26.3.4 m_points . . . 306
10.26.3.5 m_tensor . . . 306
10.26.3.6 m_width . . . 306
10.27 serif::discretization::backend::mfem::BasisWorkspace Struct Reference . . . 307
10.27.1 Detailed Description . . . 307
10.27.2 Member Data Documentation . . . 307
10.27.2.1 reference_values . . . 307
10.27.2.2 tensor . . . 307
10.28 serif::operators::backend::mfem::BasisWorkspace Struct Reference . . . 308
10.28.1 Detailed Description . . . 308
10.28.2 Member Data Documentation . . . 308
10.28.2.1 reference_values . . . 308
10.28.2.2 tensor . . . 308

10.29 serif::utils::dependencies::Binding< KeyT, TargetT > Struct Template Reference 309
10.29.1 Member Typedef Documentation 309
10.29.1.1 Key 309
10.29.1.2 Target 309
10.30 serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< K, T > Struct Template Reference 309
10.30.1 Member Typedef Documentation 309
10.30.1.1 type 309
10.31 serif::utils::dependencies::Bindings< BindingTs > Struct Template Reference 310
10.32 serif::discretization::blocks::block Struct Reference 310
10.32.1 Detailed Description 311
10.33 serif::discretization::blocks::central_density::border Struct Reference 312
10.33.1 Detailed Description 313
10.33.2 Member Typedef Documentation 313
10.33.2.1 Quantity 313
10.33.2.2 Support 313
10.33.3 Member Data Documentation 313
10.33.3.1 name 313
10.33.3.2 symbol 314
10.34 serif::operators::backend::mfem::nodal_keys::BorderLinearization Struct Reference 314
10.35 serif::utils::dependencies::detail::BorrowedValue< T > Struct Template Reference 314
10.35.1 Member Typedef Documentation 314
10.35.1.1 Target 314
10.35.2 Member Function Documentation 315
10.35.2.1 get() 315
10.35.3 Member Data Documentation 315
10.35.3.1 value 315
10.36 serif::discretization::domain::Boundary Struct Reference 315
10.37 serif::discretization::forms::boundary_flux< TrialT, TestT > Struct Template Reference . . . 316
10.37.1 Detailed Description 317
10.37.2 Member Typedef Documentation 317
10.37.2.1 operand_types 317
10.37.2.2 test_type 317
10.37.2.3 trial_type 318
10.37.3 Member Data Documentation 318
10.37.3.1 name 318
10.38 serif::discretization::domain::ids::BoundaryDescriptor Struct Reference 318
10.38.1 Detailed Description 319
10.38.2 Member Data Documentation 319
10.38.2.1 ID 319

10.38.2.2 name . . . 319
10.39 serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue > Struct Template Reference . . . 319
10.39.1 Detailed Description . . . 320
10.39.2 Member Typedef Documentation . . . 320
10.39.2.1 boundary_type . . . 320
10.39.3 Member Data Documentation . . . 320
10.39.3.1 ID . . . 320
10.40 serif::discretization::domain::schema::BoundaryID< BoundaryT, IdentifierValue > Struct Template Reference . . . 321
10.40.1 Detailed Description . . . 321
10.40.2 Member Typedef Documentation . . . 321
10.40.2.1 boundary_type . . . 321
10.40.3 Member Data Documentation . . . 322
10.40.3.1 ID . . . 322
10.41 serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs > Struct Template Reference . . . 322
10.41.1 Detailed Description . . . 322
10.41.2 Member Function Documentation . . . 323
10.41.2.1 descriptors() . . . 323
10.41.3 Member Data Documentation . . . 323
10.41.3.1 count . . . 323
10.42 serif::discretization::domain::schema::BoundaryIDList< BoundaryIDTs > Struct Template Reference . . . 323
10.42.1 Detailed Description . . . 324
10.42.2 Member Function Documentation . . . 324
10.42.2.1 descriptors() . . . 324
10.42.3 Member Data Documentation . . . 324
10.42.3.1 count . . . 324
10.43 serif::discretization::domain::BoundaryIDResolver< BoundaryT, BoundaryIDListT > Struct Template Reference . . . 324
10.44 serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > > Struct Template Reference . . . 325
10.44.1 Member Typedef Documentation . . . 325
10.44.1.1 ResolverFor . . . 325
10.44.2 Member Function Documentation . . . 325
10.44.2.1 contains_id() . . . 325
10.44.3 Member Data Documentation . . . 325
10.44.3.1 registered . . . 325
10.45 serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > > Struct Template Reference . . . 326
10.45.1 Member Function Documentation . . . 326

10.45.1.1 contains_id() . . . 326
10.45.1.2 id() . . . 326
10.45.2 Member Data Documentation . . . 326
10.45.2.1 registered . . . 326
10.46 serif::discretization::domain::BoundarySet< BoundaryTs > Struct Template Reference . . . 327
10.46.1 Member Data Documentation . . . 327
10.46.1.1 count . . . 327
10.47 serif::geometry::surface::BoundaryValues< Term > Struct Template Reference . . . 327
10.47.1 Detailed Description . . . 328
10.47.2 Member Function Documentation . . . 328
10.47.2.1 operator==() . . . 328
10.47.3 Member Data Documentation . . . 328
10.47.3.1 values . . . 328
10.48 serif::geometry::surface::detail::BoundaryValueTracking Struct Reference . . . 329
10.48.1 Member Function Documentation . . . 330
10.48.1.1 identity() . . . 330
10.49 serif::utils::dependencies::BoundType< BindingsT, KeyT > Struct Template Reference . . . 330
10.50 serif::utils::dependencies::detail::BoundType< BindingsT, KeyT > Struct Template Reference 331
10.51 serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT > Struct Template Reference . . . 331
10.51.1 Member Typedef Documentation . . . 332
10.51.1.1 target_type . . . 332
10.51.2 Member Function Documentation . . . 332
10.51.2.1 select() . . . 332
10.52 serif::utils::dependencies::detail::CachedResult< ValueT, InputCount > Struct Template Reference . . . 332
10.52.1 Member Data Documentation . . . 333
10.52.1.1 input_versions . . . 333
10.52.1.2 value . . . 333
10.53 serif::eos::relations::CatalogContainsRelation< Catalog, RelationT > Struct Template Reference333
10.54 serif::eos::relations::CatalogContainsRelation< RelationCatalog< Relations... >, RelationT > Struct Template Reference . . . 334
10.55 serif::equations::CenteringConstraint Struct Reference . . . 335
10.55.1 Detailed Description . . . 336
10.55.2 Member Typedef Documentation . . . 336
10.55.2.1 Operands . . . 336
10.55.3 Member Function Documentation . . . 337
10.55.3.1 evaluate() . . . 337
10.55.4 Member Data Documentation . . . 337
10.55.4.1 name . . . 337
10.55.4.2 position_independent . . . 337

10.55.4.3 reference_position . . . 337
10.56 serif::discretization::blocks::central_density Struct Reference . . . 337
10.56.1 Detailed Description . . . 338
10.56.2 Member Typedef Documentation . . . 338
10.56.2.1 Terms . . . 338
10.56.3 Member Data Documentation . . . 339
10.56.3.1 border_term . . . 339
10.56.3.2 name . . . 339
10.57 serif::equations::CentralDensityConstraint< EOST > Class Template Reference . . . 339
10.57.1 Detailed Description . . . 340
10.57.2 Member Typedef Documentation . . . 341
10.57.2.1 Border . . . 341
10.57.2.2 Operands . . . 341
10.57.3 Constructor & Destructor Documentation . . . 341
10.57.3.1 CentralDensityConstraint() . . . 341
10.57.4 Member Function Documentation . . . 341
10.57.4.1 density_from_enthalpy() . . . 341
10.57.4.2 evaluate() . . . 341
10.57.4.3 target_density() . . . 342
10.57.4.4 target_enthalpy() . . . 342
10.57.5 Member Data Documentation . . . 342
10.57.5.1 affine . . . 342
10.57.5.2 m_eos . . . 342
10.57.5.3 m_target_density . . . 342
10.57.5.4 m_target_enthalpy . . . 342
10.57.5.5 name . . . 342
10.57.5.6 position_independent . . . 343
10.57.5.7 reference_position . . . 343
10.58 serif::geometry::deformation::backend::mfem::volume_displacement_keys::CheckedBinding Struct Reference . . . 343
10.59 serif::dimensions::ChemicalPotential Struct Reference . . . 343
10.59.1 Detailed Description . . . 344
10.59.2 Member Data Documentation . . . 344
10.59.2.1 identifier . . . 344
10.60 serif::geometry::deformation::backend::mfem::volume_displacement_keys::Coefficients Struct Reference . . . 345
10.61 serif::discretization::blocks::surface_deformation::coefficients Struct Reference . . . 345
10.61.1 Detailed Description . . . 346
10.61.2 Member Data Documentation . . . 347
10.61.2.1 name . . . 347

10.62 serif::discretization::backend::mfem::CoefficientTracking Struct Reference . . . 347
10.62.1 Detailed Description . . . 348
10.62.2 Member Function Documentation . . . 348
10.62.2.1 capture() [1/2] . . . 348
10.62.2.2 capture() [2/2] . . . 349
10.62.2.3 identity() . . . 349
10.62.2.4 matches() . . . 350
10.63 serif::operators::backend::mfem::CoefficientTracking Struct Reference . . . 350
10.63.1 Detailed Description . . . 351
10.63.2 Member Function Documentation . . . 351
10.63.2.1 capture() [1/2] . . . 351
10.63.2.2 capture() [2/2] . . . 352
10.63.2.3 identity() . . . 352
10.63.2.4 matches() . . . 353
10.64 serif::operators::backend::mfem::CoefficientView< BindingT > Struct Template Reference . . . 353
10.64.1 Member Typedef Documentation . . . 354
10.64.1.1 Term . . . 354
10.64.2 Member Data Documentation . . . 354
10.64.2.1 binding . . . 354
10.64.2.2 values . . . 354
10.65 serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::collector Struct Reference . . . 354
10.66 serif::operators::detail::Columns< ListT > Struct Template Reference . . . 355
10.67 serif::operators::detail::Columns< utils::types::TypeList< O... > > Struct Template Reference 355
10.67.1 Member Typedef Documentation . . . 355
10.67.1.1 type . . . 355
10.68 serif::operators::backend::mfem::CompactificationCoefficients Struct Reference . . . 356
10.69 serif::mapping::CompactificationPointData Struct Reference . . . 356
10.69.1 Member Data Documentation . . . 356
10.69.1.1 compactification_coordinate . . . 356
10.69.1.2 coordinate_gradient . . . 356
10.70 serif::utils::CompensatedSum Class Reference . . . 356
10.70.1 Detailed Description . . . 357
10.70.2 Member Function Documentation . . . 357
10.70.2.1 add() . . . 357
10.70.2.2 value() . . . 357
10.70.3 Member Data Documentation . . . 357
10.70.3.1 m_correction . . . 357
10.70.3.2 m_sum . . . 357

10.71 serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap > Class Template Reference . . . 358
10.71.1 Detailed Description . . . 359
10.71.2 Member Typedef Documentation . . . 360
10.71.2.1 InputSpace . . . 360
10.71.2.2 OutputSpace . . . 360
10.71.3 Constructor & Destructor Documentation . . . 360
10.71.3.1 ComposedDeformation() . . . 360
10.71.4 Member Function Documentation . . . 360
10.71.4.1 apply_jacobian() . . . 360
10.71.4.2 apply_jacobian_transpose() . . . 360
10.71.4.3 apply_pullback_derivative() . . . 361
10.71.4.4 build_displacement() . . . 361
10.71.4.5 extension_map() . . . 361
10.71.4.6 input_size() . . . 361
10.71.4.7 make_workspace() . . . 361
10.71.4.8 output_size() . . . 362
10.71.4.9 surface_map() . . . 362
10.71.5 Friends And Related Symbol Documentation . . . 362
10.71.5.1 compose_deformation . . . 362
10.71.6 Member Data Documentation . . . 362
10.71.6.1 is_linear . . . 362
10.71.6.2 m_extension . . . 362
10.71.6.3 m_surface . . . 363
10.72 serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT > Class Template Reference . . . 363
10.72.1 Member Typedef Documentation . . . 363
10.72.1.1 Function . . . 363
10.72.1.2 Inputs . . . 364
10.72.1.3 Key . . . 364
10.72.2 Constructor & Destructor Documentation . . . 364
10.72.2.1 Computation() . . . 364
10.72.3 Member Function Documentation . . . 364
10.72.3.1 evaluate() . . . 364
10.72.4 Member Data Documentation . . . 364
10.72.4.1 m_function . . . 364
10.73 serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT > Class Template Reference . . . 365
10.73.1 Member Typedef Documentation . . . 367
10.73.1.1 Bindings . . . 367
10.73.1.2 Failure . . . 367

10.73.1.3 Report . . . 367
10.73.1.4 ResultMask . . . 367
10.73.1.5 SourceMask . . . 367
10.73.1.6 Sources . . . 367
10.73.1.7 Storage . . . 367
10.73.1.8 Traits . . . 368
10.73.2 Constructor & Destructor Documentation . . . 368
10.73.2.1 ComputationCache() [1/3] . . . 368
10.73.2.2 ComputationCache() [2/3] . . . 368
10.73.2.3 ComputationCache() [3/3] . . . 368
10.73.3 Member Function Documentation . . . 368
10.73.3.1 contains() . . . 368
10.73.3.2 get() . . . 369
10.73.3.3 input() . . . 369
10.73.3.4 inspect_source() . . . 369
10.73.3.5 invalidate() . . . 369
10.73.3.6 operator=() [1/2] . . . 369
10.73.3.7 operator=() [2/2] . . . 370
10.73.3.8 preparation_count() . . . 370
10.73.3.9 prepare_in_order() . . . 370
10.73.3.10 prepare_one() . . . 370
10.73.3.11 ready() . . . 370
10.73.3.12 report() . . . 371
10.73.3.13 try_prepare() . . . 371
10.73.3.14 version_index() . . . 371
10.73.4 Member Data Documentation . . . 371
10.73.4.1 m_active . . . 371
10.73.4.2 m_graph . . . 371
10.73.4.3 m_preparation_count . . . 371
10.73.4.4 m_ready . . . 372
10.73.4.5 m_report . . . 372
10.73.4.6 m_storage . . . 372
10.73.4.7 source_count . . . 372
10.74 serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, ResourcesT, IndicesT > Struct Template Reference . . . 372
10.75 serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > > Struct Template Reference 373
10.75.1 Member Data Documentation . . . 373
10.75.1.1 results . . . 373
10.75.1.2 sources . . . 374

10.75.1.3 versions . . . 374
10.76 serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, Valid > Struct Template Reference . . . 374
10.76.1 Member Data Documentation . . . 374
10.76.1.1 valid . . . 374
10.77 serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true > Struct Template Reference . . . 374
10.77.1 Member Typedef Documentation . . . 375
10.77.1.1 Bindings . . . 375
10.77.1.2 Resources . . . 375
10.77.1.3 Sources . . . 375
10.77.2 Member Data Documentation . . . 375
10.77.2.1 valid [1/2] . . . 375
10.77.2.2 valid [2/2] . . . 375
10.78 serif::utils::dependencies::ComputationGraph< Computations > Class Template Reference . . . 376
10.78.1 Member Typedef Documentation . . . 376
10.78.1.1 ComputationAt . . . 376
10.78.1.2 Dependencies . . . 377
10.78.1.3 Inputs . . . 377
10.78.1.4 Results . . . 377
10.78.2 Constructor & Destructor Documentation . . . 377
10.78.2.1 ComputationGraph() . . . 377
10.78.3 Member Function Documentation . . . 377
10.78.3.1 computation() . . . 377
10.78.3.2 index() . . . 377
10.78.4 Member Data Documentation . . . 378
10.78.4.1 m_computations . . . 378
10.78.4.2 size . . . 378
10.79 serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, Independent > Struct Template Reference . . . 378
10.79.1 Member Data Documentation . . . 378
10.79.1.1 valid . . . 378
10.80 serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, true > Struct Template Reference . . . 379
10.80.1 Member Data Documentation . . . 379
10.80.1.1 valid . . . 379
10.81 serif::utils::dependencies::ComputationStore< EntryTs > Class Template Reference . . . 380
10.81.1 Detailed Description . . . 380
10.81.2 Member Typedef Documentation . . . 381
10.81.2.1 Bindings . . . 381
10.81.3 Constructor & Destructor Documentation . . . 381

10.81.3.1 ComputationStore() [1/3] . . . 381
10.81.3.2 ComputationStore() [2/3] . . . 381
10.81.3.3 ComputationStore() [3/3] . . . 381
10.81.4 Member Function Documentation . . . 381
10.81.4.1 get() [1/4] . . . 381
10.81.4.2 get() [2/4] . . . 382
10.81.4.3 get() [3/4] . . . 382
10.81.4.4 get() [4/4] . . . 382
10.81.4.5 operator=() [1/2] . . . 382
10.81.4.6 operator=() [2/2] . . . 382
10.81.5 Member Data Documentation . . . 382
10.81.5.1 m_entries . . . 382
10.82 serif::utils::dependencies::ComputationUpdate< InputsT, ValueT > Struct Template Reference 383
10.82.1 Member Function Documentation . . . 383
10.82.1.1 changed() . . . 383
10.82.2 Member Data Documentation . . . 383
10.82.2.1 changed_inputs . . . 383
10.82.2.2 previous . . . 384
10.82.2.3 rebuild . . . 384
10.83 serif::utils::misc::concepts::Concat< Lists > Struct Template Reference . . . 384
10.84 serif::utils::misc::concepts::Concat< types::TypeList< LhsTs... >, types::TypeList< RhsTs... >, Rest... > Struct Template Reference . . . 384
10.84.1 Member Typedef Documentation . . . 384
10.84.1.1 Type . . . 384
10.85 serif::utils::misc::concepts::Concat< types::TypeList< Ts... > > Struct Template Reference . 385
10.85.1 Member Typedef Documentation . . . 385
10.85.1.1 Type . . . 385
10.86 serif::utils::misc::concepts::Concat<> Struct Reference . . . 385
10.86.1 Member Typedef Documentation . . . 385
10.86.1.1 Type . . . 385
10.87 serif::discretization::domain::relation::validation::RelationValidationResult::Connected↩ Diagnostics Struct Reference . . . 386
10.87.1 Member Data Documentation . . . 386
10.87.1.1 domainElementCount . . . 386
10.87.1.2 elementID . . . 386
10.87.1.3 visitedElementCount . . . 386
10.88 serif::discretization::domain::schema::validation::RelationValidationResult::Connected↩ Diagnostics Struct Reference . . . 386
10.88.1 Member Data Documentation . . . 387
10.88.1.1 domainElementCount . . . 387
10.88.1.2 elementID . . . 387

10.88.1.3 visitedElementCount . . . 387
10.89 serif::equations::Contribution< TestOperandT, SupportT > Struct Template Reference . . . 387
10.89.1 Detailed Description . . . 387
10.89.2 Member Typedef Documentation . . . 388
10.89.2.1 Support . . . 388
10.89.2.2 Test . . . 388
10.89.2.3 TestOperand . . . 388
10.90 serif::operators::ContributionBinding< EquationT > Struct Template Reference . . . 388
10.90.1 Detailed Description . . . 389
10.90.2 Member Data Documentation . . . 389
10.90.2.1 coefficient . . . 389
10.90.2.2 equation . . . 389
10.91 serif::operators::detail::ContributionChildren< EquationT > Struct Template Reference . . . 389
10.91.1 Detailed Description . . . 390
10.91.2 Constructor & Destructor Documentation . . . 390
10.91.2.1 ContributionChildren() . . . 390
10.91.3 Member Data Documentation . . . 390
10.91.3.1 parent . . . 390
10.91.3.2 values . . . 390
10.92 serif::operators::backend::mfem::ContributionData< OperandCount > Struct Template Reference . . . 391
10.92.1 Member Data Documentation . . . 391
10.92.1.1 elements . . . 391
10.92.1.2 residual_offset . . . 391
10.93 serif::operators::detail::ContributionOperands< TupleT > Struct Template Reference . . . 391
10.94 serif::operators::detail::ContributionOperands< std::tuple< C... > > Struct Template Reference 392
10.95 serif::equations::Contributions< T, typename > Struct Template Reference . . . 393
10.95.1 Detailed Description . . . 393
10.95.2 Member Typedef Documentation . . . 393
10.95.2.1 type . . . 393
10.96 serif::equations::Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions())> > Struct Template Reference . . . 393
10.96.1 Member Typedef Documentation . . . 394
10.96.1.1 type . . . 394
10.97 serif::equations::ContributionSupport< T, typename > Struct Template Reference . . . 394
10.97.1 Member Typedef Documentation . . . 395
10.97.1.1 type . . . 395
10.98 serif::equations::ContributionSupport< T, std::void_t< typename T::Support > > Struct Template Reference . . . 395
10.98.1 Member Typedef Documentation . . . 395
10.98.1.1 type [1/2] . . . 395

10.98.1.2 type [2/2] . . . 395
10.99 serif::equations::ContributionTuple< TupleT, typename > Struct Template Reference . . . 395
10.99.1 Member Typedef Documentation . . . 396
10.99.1.1 type . . . 396
10.100 serif::equations::ContributionTuple< std::tuple< Ts... >, std::void_t< decltype(std::tuple_cat(std::declval< typename Contributions< std::remove_cvref_t< Ts > >::type >()...))> > Struct Template Reference . . . 396
10.100.1 Member Typedef Documentation . . . 396
10.100.1.1 type [1/2] . . . 396
10.100.1.2 type [2/2] . . . 396
10.101 serif::utils::dependencies::keys::Coordinates Struct Reference . . . 397
10.101.1 Detailed Description . . . 397
10.102 serif::discretization::domain::CoreDomain Struct Reference . . . 397
10.102.1 Member Data Documentation . . . 398
10.102.1.1 name . . . 398
10.103 serif::operators::detail::CouplingForm< Input, TestOperand > Struct Template Reference . . . 398
10.103.1 Member Typedef Documentation . . . 399
10.103.1.1 Test . . . 399
10.103.1.2 Trial . . . 399
10.103.1.3 type . . . 399
10.104 serif::utils::types::ct_error_string< N > Struct Template Reference . . . 399
10.104.1 Constructor & Destructor Documentation . . . 400
10.104.1.1 ct_error_string() . . . 400
10.104.2 Member Function Documentation . . . 400
10.104.2.1 view() . . . 400
10.104.3 Member Data Documentation . . . 400
10.104.3.1 data . . . 400
10.104.3.2 padding . . . 400
10.105 serif::utils::types::ctstring< N > Struct Template Reference . . . 400
10.105.1 Constructor & Destructor Documentation . . . 401
10.105.1.1 ctstring() . . . 401
10.105.2 Member Function Documentation . . . 401
10.105.2.1 view() . . . 401
10.105.3 Member Data Documentation . . . 401
10.105.3.1 data . . . 401
10.106 serif::discretization::forms::Operation::Curl Struct Reference . . . 401
10.106.1 Detailed Description . . . 402
10.107 serif::geometry::deformation::DeformationFailure Struct Reference . . . 402
10.107.1 Detailed Description . . . 402
10.107.2 Member Data Documentation . . . 403

10.107.2.1 element . . . 403
10.107.2.2 message . . . 403
10.107.2.3 point . . . 403
10.107.2.4 reason . . . 403
10.108 serif::geometry::deformation::DeformationLinearization< Map > Struct Template Reference 403
10.108.1 Detailed Description . . . 404
10.108.2 Member Data Documentation . . . 404
10.108.2.1 map . . . 404
10.108.2.2 parameters . . . 404
10.109 serif::geometry::deformation::backend::mfem::DeformationValueTracking Struct Reference . 404
10.109.1 Detailed Description . . . 406
10.109.2 Member Function Documentation . . . 406
10.109.2.1 identity() . . . 406
10.110 serif::geometry::deformation::DeformationWorkspaceType< Map, Linear > Struct Template Reference . . . 406
10.110.1 Detailed Description . . . 406
10.110.2 Member Typedef Documentation . . . 407
10.110.2.1 Type . . . 407
10.111 serif::geometry::deformation::DeformationWorkspaceType< Map, true > Struct Template Reference . . . 407
10.111.1 Member Typedef Documentation . . . 407
10.111.1.1 Type [1/2] . . . 407
10.111.1.2 Type [2/2] . . . 407
10.112 serif::utils::types::DenseMatrixBase< T > Struct Template Reference . . . 407
10.112.1 Member Typedef Documentation . . . 408
10.112.1.1 Type . . . 408
10.113 serif::utils::types::DenseTensorBase< T, Rank > Struct Template Reference . . . 408
10.113.1 Member Typedef Documentation . . . 408
10.113.1.1 Type . . . 408
10.113.2 Member Data Documentation . . . 408
10.113.2.1 rank . . . 408
10.114 serif::discretization::blocks::density Struct Reference . . . 409
10.114.1 Detailed Description . . . 410
10.114.2 Member Typedef Documentation . . . 410
10.114.2.1 Terms . . . 410
10.114.3 Member Data Documentation . . . 410
10.114.3.1 mass_density_term . . . 410
10.114.3.2 name . . . 410
10.115 serif::equations::gravity_contributions::DensitySource Struct Reference . . . 411
10.115.1 Member Typedef Documentation . . . 412

10.115.1.1 Operands . . . 412
10.115.2 Constructor & Destructor Documentation . . . 412
10.115.2.1 DensitySource() . . . 412
10.115.3 Member Function Documentation . . . 412
10.115.3.1 additional_order() . . . 412
10.115.3.2 evaluate() . . . 412
10.115.4 Member Data Documentation . . . 412
10.115.4.1 four_pi_G . . . 412
10.116 serif::utils::dependencies::DependencyContext< ResourceTs > Class Template Reference . . . 413
10.116.1 Member Typedef Documentation . . . 413
10.116.1.1 Bindings . . . 413
10.116.1.2 ResourceBindings . . . 414
10.116.1.3 Resources . . . 414
10.116.2 Constructor & Destructor Documentation . . . 414
10.116.2.1 DependencyContext() . . . 414
10.116.3 Member Function Documentation . . . 414
10.116.3.1 get() . . . 414
10.116.3.2 resource() . . . 414
10.116.3.3 select() [1/2] . . . 415
10.116.3.4 select() [2/2] . . . 415
10.116.4 Member Data Documentation . . . 415
10.116.4.1 m_resources . . . 415
10.117 serif::utils::dependencies::DependencyContextView< ParentT, KeysT > Class Template Reference . . . 415
10.118 serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > > Class Template Reference . . . 415
10.118.1 Member Typedef Documentation . . . 416
10.118.1.1 Bindings . . . 416
10.118.1.2 ResourceBindings . . . 416
10.118.1.3 Resources . . . 416
10.118.2 Constructor & Destructor Documentation . . . 417
10.118.2.1 DependencyContextView() [1/2] . . . 417
10.118.2.2 DependencyContextView() [2/2] . . . 417
10.118.3 Member Function Documentation . . . 417
10.118.3.1 get() . . . 417
10.118.3.2 resource() . . . 417
10.118.4 Member Data Documentation . . . 417
10.118.4.1 m_parent . . . 417
10.119 serif::dimensions::Dimensionless Struct Reference . . . 418
10.119.1 Detailed Description . . . 419

10.119.2 Member Data Documentation . . . 419
10.119.2.1 identifier . . . 419
10.120 serif::geometry::surface::DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation > Struct Template Reference . . . 419
10.120.1 Detailed Description . . . 420
10.121 serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > Struct Template Reference . . . 421
10.121.1 Detailed Description . . . 421
10.121.2 Member Typedef Documentation . . . 422
10.121.2.1 rank . . . 422
10.121.3 Member Data Documentation . . . 422
10.121.3.1 functional_space . . . 422
10.121.3.2 order . . . 422
10.122 serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< FunctionalSpaceT > Struct Template Reference . . . 422
10.123 serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > > Struct Template Reference . . . 423
10.123.1 Member Data Documentation . . . 423
10.123.1.1 order . . . 423
10.123.1.2 rank . . . 423
10.123.1.3 space . . . 423
10.124 serif::discretization::blocks::displacement Struct Reference . . . 424
10.124.1 Detailed Description . . . 425
10.124.2 Member Typedef Documentation . . . 425
10.124.2.1 Terms . . . 425
10.124.3 Member Data Documentation . . . 425
10.124.3.1 name . . . 425
10.124.3.2 vector_term . . . 425
10.125 serif::operators::backend::mfem::DisplacementCoefficients Struct Reference . . . 425
10.126 serif::equations::DisplacementForce< EOST > Class Template Reference . . . 426
10.126.1 Detailed Description . . . 426
10.126.2 Member Typedef Documentation . . . 427
10.126.2.1 Test . . . 427
10.126.3 Constructor & Destructor Documentation . . . 427
10.126.3.1 DisplacementForce() . . . 427
10.126.4 Member Function Documentation . . . 427
10.126.4.1 contributions() . . . 427
10.126.5 Member Data Documentation . . . 427
10.126.5.1 m_equation . . . 427
10.126.5.2 name . . . 427
10.127 serif::discretization::forms::Operation::Divergence Struct Reference . . . 428

10.127.1 Detailed Description . . . 428
10.128 serif::discretization::forms::divergence< TrialT, TestT > Struct Template Reference . . . 428
10.128.1 Detailed Description . . . 429
10.128.2 Member Typedef Documentation . . . 430
10.128.2.1 operand_types . . . 430
10.128.2.2 test_type . . . 430
10.128.2.3 trial_type . . . 430
10.128.3 Member Data Documentation . . . 430
10.128.3.1 name . . . 430
10.129 serif::mapping::DMFixedMeshGeometry Struct Reference . . . 431
10.129.1 Member Data Documentation . . . 431
10.129.1.1 inverse_jacobian . . . 431
10.129.1.2 jacobian . . . 431
10.129.1.3 position . . . 431
10.129.1.4 weight . . . 431
10.130 serif::mapping::DMGeometryCache Struct Reference . . . 431
10.130.1 Member Data Documentation . . . 432
10.130.1.1 enabled . . . 432
10.130.1.2 hits . . . 432
10.130.1.3 mesh . . . 432
10.130.1.4 misses . . . 432
10.130.1.5 nodesSequence . . . 432
10.130.1.6 points . . . 432
10.130.1.7 sequence . . . 432
10.131 serif::mapping::DMGeometryCacheStatistics Struct Reference . . . 433
10.131.1 Member Data Documentation . . . 433
10.131.1.1 hits . . . 433
10.131.1.2 misses . . . 433
10.131.1.3 points . . . 433
10.132 serif::mapping::DMReferenceDisplacementSample Struct Reference . . . 433
10.132.1 Member Data Documentation . . . 433
10.132.1.1 gradient . . . 433
10.132.1.2 value . . . 434
10.133 serif::mapping::DMWorkspace Class Reference . . . 434
10.133.1 Constructor & Destructor Documentation . . . 435
10.133.1.1 DMWorkspace() . . . 435
10.133.2 Member Function Documentation . . . 435
10.133.2.1 get_dimension() . . . 435
10.133.2.2 set_dimension() . . . 435
10.133.3 Member Data Documentation . . . 435

10.133.3.1 m_compactification_dshape . . . 435
10.133.3.2 m_compactification_point . . . 435
10.133.3.3 m_compactification_shape . . . 435
10.133.3.4 m_dimension . . . 435
10.133.3.5 m_exterior_result . . . 436
10.133.3.6 m_exterior_variation . . . 436
10.133.3.7 m_field_jacobian . . . 436
10.133.3.8 m_field_value . . . 436
10.133.3.9 m_full_element_jacobian . . . 436
10.133.3.10 m_mapped_normal . . . 436
10.133.3.11 m_matrix_temp_1 . . . 436
10.133.3.12 m_matrix_temp_2 . . . 436
10.133.3.13 m_mesh_dshape . . . 436
10.133.3.14 m_reference_dshape . . . 436
10.133.3.15 m_reference_field_jacobian . . . 437
10.133.3.16 m_reference_normal . . . 437
10.133.3.17 m_shape . . . 437
10.133.3.18 m_vector_temp . . . 437
10.134 serif::discretization::backend::mfem::DofMap Class Reference . . . 437
10.134.1 Detailed Description . . . 440
10.134.2 Constructor & Destructor Documentation . . . 440
10.134.2.1 DofMap() [1/4] . . . 440
10.134.2.2 DofMap() [2/4] . . . 440
10.134.2.3 ~DofMap() . . . 440
10.134.2.4 DofMap() [3/4] . . . 440
10.134.2.5 DofMap() [4/4] . . . 440
10.134.3 Member Function Documentation . . . 440
10.134.3.1 add_element() . . . 440
10.134.3.2 element_values() . . . 441
10.134.3.3 full_size() . . . 441
10.134.3.4 gather() . . . 441
10.134.3.5 mesh() . . . 442
10.134.3.6 operator=() [1/2] . . . 442
10.134.3.7 operator=() [2/2] . . . 442
10.134.3.8 reduced_dof() . . . 442
10.134.3.9 scatter() . . . 442
10.134.3.10 size() . . . 443
10.134.3.11 space() . . . 443
10.134.3.12 supports() . . . 443
10.134.3.13 validate() . . . 443

10.134.3.14 validate_values() . . . 443
10.134.4 Member Data Documentation . . . 444
10.134.4.1 m_attributes . . . 444
10.134.4.2 m_element_support . . . 444
10.134.4.3 m_full_to_reduced . . . 444
10.134.4.4 m_mesh . . . 444
10.134.4.5 m_mesh_sequence . . . 444
10.134.4.6 m_reduced_to_full . . . 444
10.134.4.7 m_space . . . 445
10.134.4.8 m_space_sequence . . . 445
10.135 serif::discretization::backend::mfem::DofSelection Class Reference . . . 445
10.135.1 Detailed Description . . . 446
10.135.2 Constructor & Destructor Documentation . . . 446
10.135.2.1 DofSelection() . . . 446
10.135.3 Member Function Documentation . . . 447
10.135.3.1 binding() . . . 447
10.135.3.2 boundary_attributes() . . . 447
10.135.3.3 reduced_dofs() . . . 447
10.135.3.4 size() . . . 448
10.135.3.5 tracks_coordinates() . . . 448
10.135.3.6 validate() . . . 448
10.135.4 Member Data Documentation . . . 448
10.135.4.1 m_binding . . . 448
10.135.4.2 m_boundary_attributes . . . 448
10.135.4.3 m_dofs . . . 449
10.135.4.4 m_nodes_sequence . . . 449
10.136 serif::discretization::domain::Domain Struct Reference . . . 449
10.137 serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs > Struct Template Reference . . . 450
10.137.1 Member Typedef Documentation . . . 451
10.137.1.1 boundary_type . . . 451
10.137.1.2 domain_types . . . 451
10.137.2 Member Data Documentation . . . 451
10.137.2.1 domain_count . . . 451
10.137.2.2 name . . . 451
10.138 serif::discretization::domain::schema::validation::DomainBoundary< BoundaryT, DomainTs > Struct Template Reference . . . 452
10.138.1 Member Typedef Documentation . . . 453
10.138.1.1 boundary_type . . . 453
10.138.1.2 domain_types . . . 453

10.138.2 Member Data Documentation . . . 453
10.138.2.1 domain_count . . . 453
10.138.2.2 name . . . 453
10.139 serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics Struct Reference . . . 454
10.139.1 Member Data Documentation . . . 454
10.139.1.1 actualBoundaryID . . . 454
10.139.1.2 boundaryElementID . . . 454
10.139.1.3 expectedBoundaryID . . . 454
10.139.1.4 faceID . . . 454
10.139.1.5 firstDomainID . . . 454
10.139.1.6 firstElementID . . . 455
10.139.1.7 secondDomainID . . . 455
10.139.1.8 secondElementID . . . 455
10.140 serif::discretization::domain::schema::validation::RelationValidationResult::DomainBoundaryDiagnostics Struct Reference . . . 455
10.140.1 Member Data Documentation . . . 455
10.140.1.1 actualBoundaryID . . . 455
10.140.1.2 boundaryElementID . . . 456
10.140.1.3 expectedBoundaryID . . . 456
10.140.1.4 faceID . . . 456
10.140.1.5 firstDomainID . . . 456
10.140.1.6 firstElementID . . . 456
10.140.1.7 secondDomainID . . . 456
10.140.1.8 secondElementID . . . 456
10.141 serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT > Struct Template Reference . . . 457
10.141.1 Detailed Description . . . 457
10.141.2 Member Function Documentation . . . 457
10.141.2.1 operator==() . . . 457
10.141.3 Member Data Documentation . . . 458
10.141.3.1 stellar_interior . . . 458
10.141.3.2 surface . . . 458
10.141.3.3 vacuum . . . 458
10.142 serif::discretization::domain::ids::DomainDescriptor Struct Reference . . . 458
10.142.1 Detailed Description . . . 459
10.142.2 Member Data Documentation . . . 459
10.142.2.1 ID . . . 459
10.142.2.2 name . . . 460
10.143 serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue > Struct Template Reference . . . 460

10.143.1 Detailed Description . . . 460
10.143.2 Member Typedef Documentation . . . 461
10.143.2.1 domain_type . . . 461
10.143.3 Member Data Documentation . . . 461
10.143.3.1 ID . . . 461
10.144 serif::discretization::domain::schema::DomainID< DomainT, IdentifierValue > Struct Template Reference . . . 461
10.144.1 Detailed Description . . . 461
10.144.2 Member Typedef Documentation . . . 462
10.144.2.1 domain_type . . . 462
10.144.3 Member Data Documentation . . . 462
10.144.3.1 ID . . . 462
10.145 serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs > Struct Template Reference . . . 462
10.145.1 Detailed Description . . . 463
10.145.2 Member Function Documentation . . . 463
10.145.2.1 descriptors() . . . 463
10.145.3 Member Data Documentation . . . 463
10.145.3.1 count . . . 463
10.146 serif::discretization::domain::schema::DomainIDList< DomainIDTs > Struct Template Reference . . . 463
10.146.1 Detailed Description . . . 464
10.146.2 Member Function Documentation . . . 464
10.146.2.1 descriptors() . . . 464
10.146.3 Member Data Documentation . . . 464
10.146.3.1 count . . . 464
10.147 serif::discretization::domain::DomainIDResolver< DomainT, DomainIDListT > Struct Template Reference . . . 465
10.148 serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... > > Struct Template Reference . . . 465
10.148.1 Member Typedef Documentation . . . 465
10.148.1.1 ResolverFor . . . 465
10.148.2 Member Function Documentation . . . 465
10.148.2.1 contains_id() . . . 465
10.148.3 Member Data Documentation . . . 466
10.148.3.1 registered . . . 466
10.149 serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > > Struct Template Reference . . . 466
10.149.1 Member Function Documentation . . . 466
10.149.1.1 contains_id() . . . 466
10.149.1.2 id() . . . 466

10.149.2 Member Data Documentation . . . 467
10.149.2.1 registered . . . 467
10.150 serif::geometry::deformation::keys::DomainMap Struct Reference . . . 467
10.151 serif::mapping::DomainMapper Class Reference . . . 467
10.151.1 Constructor & Destructor Documentation . . . 469
10.151.1.1 DomainMapper() [1/3] . . . 469
10.151.1.2 ~DomainMapper() . . . 469
10.151.1.3 DomainMapper() [2/3] . . . 469
10.151.1.4 DomainMapper() [3/3] . . . 469
10.151.2 Member Function Documentation . . . 469
10.151.2.1 clear_geometry_cache() . . . 469
10.151.2.2 clone() . . . 470
10.151.2.3 evaluate_compactification_coordinate() . . . 470
10.151.2.4 evaluate_displacement_field() . . . 470
10.151.2.5 evaluate_face() . . . 470
10.151.2.6 evaluate_face_variation() . . . 470
10.151.2.7 evaluate_point() . . . 471
10.151.2.8 evaluate_point_impl() . . . 471
10.151.2.9 evaluate_point_variation() . . . 471
10.151.2.10 evaluate_point_variation_impl() . . . 471
10.151.2.11 evaluate_volume() . . . 472
10.151.2.12 evaluate_volume_variation() . . . 472
10.151.2.13 get_dimension() . . . 472
10.151.2.14 get_exterior_map() . . . 472
10.151.2.15 get_fixed_mesh_geometry() . . . 472
10.151.2.16 get_geometry_cache_statistics() . . . 472
10.151.2.17 is_compactified_element() . . . 473
10.151.2.18 is_geometry_cache_enabled() . . . 473
10.151.2.19 operator=() [1/2] . . . 473
10.151.2.20 operator=() [2/2] . . . 473
10.151.2.21 select_face_element_integration_point() . . . 473
10.151.2.22 select_face_element_transformation() . . . 473
10.151.2.23 set_geometry_cache_enabled() . . . 473
10.151.2.24 validate_element_data() . . . 473
10.151.3 Member Data Documentation . . . 474
10.151.3.1 m_exterior_map . . . 474
10.151.3.2 m_geometryCache . . . 474
10.151.3.3 m_options . . . 474
10.152 serif::utils::user_options::DomainMapperOptions Struct Reference . . . 474
10.152.1 Member Data Documentation . . . 474

10.152.1.1 dimension . . . 474
10.152.1.2 vacuum_element_attribute_ID . . . 474
10.153 serif::discretization::domain::DomainOrSet Class Reference . . . 475
10.153.1 Constructor & Destructor Documentation . . . 475
10.153.1.1 DomainOrSet() [1/2] . . . 475
10.153.1.2 DomainOrSet() [2/2] . . . 476
10.153.2 Member Function Documentation . . . 476
10.153.2.1 domains() . . . 476
10.153.3 Member Data Documentation . . . 476
10.153.3.1 m_domains . . . 476
10.154 serif::discretization::domain::DomainOrSetList< DomainTs > Struct Template Reference . . . 476
10.154.1 Member Data Documentation . . . 476
10.154.1.1 count . . . 476
10.155 serif::discretization::domain::relation::DomainRelation Struct Reference . . . 477
10.156 serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations > Struct Template Reference . . . 477
10.156.1 Member Typedef Documentation . . . 478
10.156.1.1 boundary_ids_type . . . 478
10.156.1.2 domain_ids_type . . . 478
10.156.1.3 relations_type . . . 478
10.156.2 Member Function Documentation . . . 479
10.156.2.1 boundary_descriptors() . . . 479
10.156.2.2 boundary_id() . . . 479
10.156.2.3 boundary_id_matches() . . . 479
10.156.2.4 contains_boundary() . . . 479
10.156.2.5 contains_domain() . . . 479
10.156.2.6 domain_descriptors() . . . 479
10.156.2.7 domain_id() . . . 480
10.156.2.8 domain_id_belongs_to() . . . 480
10.156.3 Member Data Documentation . . . 480
10.156.3.1 boundary_id_count . . . 480
10.156.3.2 domain_id_count . . . 480
10.156.3.3 relation_count . . . 480
10.157 serif::discretization::domain::DomainSet< DomainTs > Struct Template Reference . . . 481
10.157.1 Member Data Documentation . . . 481
10.157.1.1 count . . . 481
10.157.1.2 names . . . 482
10.158 serif::dimensions::DynamicViscosity Struct Reference . . . 482
10.158.1 Detailed Description . . . 483
10.158.2 Member Data Documentation . . . 483

10.158.2.1 identifier . . . 483
10.159 serif::operators::backend::mfem::ElementBasis< OperandCount > Struct Template Reference 484
10.159.1 Detailed Description . . . 484
10.159.2 Member Data Documentation . . . 484
10.159.2.1 displacement_basis . . . 484
10.159.2.2 element . . . 484
10.159.2.3 reference_operand_basis . . . 484
10.159.2.4 reference_test_basis . . . 485
10.159.2.5 rule . . . 485
10.160 serif::mapping::ElementCompactificationData Class Reference . . . 485
10.160.1 Constructor & Destructor Documentation . . . 485
10.160.1.1 ElementCompactificationData() . . . 485
10.160.2 Member Function Documentation . . . 485
10.160.2.1 get_dof_count() . . . 485
10.160.2.2 get_dofs() . . . 486
10.160.2.3 get_element() . . . 486
10.160.3 Member Data Documentation . . . 486
10.160.3.1 m_dofs . . . 486
10.160.3.2 m_element . . . 486
10.161 serif::operators::backend::mfem::ElementData< OperandCount > Struct Template Reference 486
10.161.1 Detailed Description . . . 487
10.161.2 Member Data Documentation . . . 487
10.161.2.1 basis . . . 487
10.161.2.2 element . . . 487
10.161.2.3 geometry . . . 487
10.161.2.4 reference_operand_values . . . 487
10.161.2.5 validation_integral . . . 488
10.161.2.6 weighted_field_derivatives . . . 488
10.161.2.7 weighted_position_derivative . . . 488
10.161.2.8 weighted_residual . . . 488
10.162 serif::mapping::ElementDisplacementData Class Reference . . . 488
10.162.1 Constructor & Destructor Documentation . . . 489
10.162.1.1 ElementDisplacementData() . . . 489
10.162.2 Member Function Documentation . . . 489
10.162.2.1 get_dimension() . . . 489
10.162.2.2 get_dof_count() . . . 489
10.162.2.3 get_dof_matrix() . . . 489
10.162.2.4 get_element() . . . 489
10.162.2.5 get_ordering() . . . 489
10.162.3 Member Data Documentation . . . 489

10.162.3.1 m_dimension . . . 489
10.162.3.2 m_dof_matrix . . . 489
10.162.3.3 m_element . . . 490
10.162.3.4 m_ordering . . . 490
10.163 serif::operators::backend::mfem::ElementGeometry< OperandCount > Struct Template Reference . . . 490
10.163.1 Detailed Description . . . 490
10.163.2 Member Data Documentation . . . 491
10.163.2.1 compactification . . . 491
10.163.2.2 displacement . . . 491
10.163.2.3 mapping . . . 491
10.163.2.4 operand_basis . . . 491
10.163.2.5 test_basis . . . 491
10.164 serif::mapping::ElementMappingData Struct Reference . . . 491
10.164.1 Member Data Documentation . . . 492
10.164.1.1 compactification . . . 492
10.164.1.2 displacement . . . 492
10.165 serif::geometry::deformation::EmptyDeformationWorkspace Struct Reference . . . 492
10.165.1 Detailed Description . . . 492
10.166 serif::dimensions::Energy Struct Reference . . . 493
10.166.1 Detailed Description . . . 494
10.166.2 Member Data Documentation . . . 494
10.166.2.1 identifier . . . 494
10.167 serif::dimensions::EnergyDensity Struct Reference . . . 494
10.167.1 Detailed Description . . . 495
10.167.2 Member Data Documentation . . . 495
10.167.2.1 identifier . . . 495
10.168 serif::discretization::blocks::enthalpy Struct Reference . . . 496
10.168.1 Detailed Description . . . 497
10.168.2 Member Typedef Documentation . . . 497
10.168.2.1 Terms . . . 497
10.168.3 Member Data Documentation . . . 497
10.168.3.1 name . . . 497
10.168.3.2 specific_term . . . 497
10.169 serif::dimensions::Entropy Struct Reference . . . 498
10.169.1 Detailed Description . . . 499
10.169.2 Member Data Documentation . . . 499
10.169.2.1 identifier . . . 499
10.170 serif::discretization::backend::mfem::TensorBasisCache::Entry Struct Reference . . . 499
10.170.1 Detailed Description . . . 500

10.170.2 Member Data Documentation . . . 500
10.170.2.1 basis . . . 500
10.170.2.2 copies . . . 500
10.170.2.3 element . . . 500
10.170.2.4 operation . . . 500
10.170.2.5 points . . . 500
10.171 serif::operators::backend::mfem::TensorBasisCache::Entry Struct Reference . . . 501
10.171.1 Detailed Description . . . 501
10.171.2 Member Data Documentation . . . 501
10.171.2.1 basis . . . 501
10.171.2.2 copies . . . 501
10.171.2.3 element . . . 501
10.171.2.4 operation . . . 502
10.171.2.5 points . . . 502
10.172 serif::utils::misc::concepts::EnumerateTypes< TypeList, Descriptor > Struct Template Reference . . . 502
10.173 serif::utils::misc::concepts::EnumerateTypes< types::TypeList< TypeTs... >, Descriptor > Struct Template Reference . . . 502
10.173.1 Member Typedef Documentation . . . 502
10.173.1.1 Type . . . 502
10.174 serif::discretization::domain::EnvelopeDomain Struct Reference . . . 503
10.174.1 Member Data Documentation . . . 504
10.174.1.1 name . . . 504
10.175 serif::eos::EOSEvaluationError Class Reference . . . 504
10.175.1 Constructor & Destructor Documentation . . . 505
10.175.1.1 EOSEvaluationError() . . . 505
10.176 serif::geometry::surface::keys::EOSResponse Struct Reference . . . 505
10.177 serif::eos::runtime::EOSView Class Reference . . . 506
10.177.1 Member Typedef Documentation . . . 507
10.177.1.1 RuntimeEvaluateFunction . . . 507
10.177.1.2 RuntimePartialDerivativeFunction . . . 507
10.177.2 Constructor & Destructor Documentation . . . 507
10.177.2.1 EOSView() [1/2] . . . 507
10.177.2.2 EOSView() [2/2] . . . 508
10.177.3 Member Function Documentation . . . 508
10.177.3.1 find_relation() . . . 508
10.177.3.2 relations() . . . 508
10.177.3.3 runtime_failure() . . . 508
10.177.3.4 supports() . . . 508
10.177.3.5 try_evaluate() [1/2] . . . 508

10.177.3.6 try_evaluate() [2/2] . . . 509
10.177.3.7 try_partial_derivative() [1/2] . . . 509
10.177.3.8 try_partial_derivative() [2/2] . . . 509
10.177.3.9 validate_relation_request() . . . 509
10.177.4 Member Data Documentation . . . 509
10.177.4.1 m_evaluate . . . 509
10.177.4.2 m_object . . . 509
10.177.4.3 m_partialDerivative . . . 509
10.177.4.4 m_relations . . . 510
10.178 serif::equations::EquationDifference< LeftT, RightT > Class Template Reference . . . 510
10.178.1 Detailed Description . . . 511
10.178.2 Member Typedef Documentation . . . 511
10.178.2.1 Test . . . 511
10.178.3 Constructor & Destructor Documentation . . . 511
10.178.3.1 EquationDifference() . . . 511
10.178.4 Member Function Documentation . . . 511
10.178.4.1 contributions() . . . 511
10.178.5 Member Data Documentation . . . 511
10.178.5.1 m_equation . . . 511
10.178.5.2 name . . . 512
10.179 serif::geometry::surface::keys::EquationOfState Struct Reference . . . 512
10.180 serif::operators::EquationPlan< EquationT, OperandsT > Struct Template Reference . . . 512
10.181 serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > > Struct Template Reference . . . 512
10.181.1 Member Typedef Documentation . . . 513
10.181.1.1 Columns . . . 513
10.181.1.2 Form . . . 513
10.181.1.3 Operands . . . 513
10.181.1.4 Support . . . 514
10.181.1.5 Test . . . 514
10.181.2 Member Function Documentation . . . 514
10.181.2.1 column_index() . . . 514
10.181.2.2 component_offset() . . . 514
10.181.2.3 for_each_column() . . . 514
10.181.2.4 for_each_input() . . . 514
10.181.2.5 for_each_operand() . . . 515
10.181.3 Member Data Documentation . . . 515
10.181.3.1 column_count . . . 515
10.181.3.2 component_count . . . 515
10.181.3.3 operand_count . . . 515

10.182 serif::equations::EquationPoint< OperandsT > Struct Template Reference 515
10.182.1 Member Data Documentation 516
10.182.1.1 fields 516
10.182.1.2 position 516
10.183 serif::equations::EquationRegistry< EquationTs > Class Template Reference 516
10.183.1 Member Typedef Documentation 517
10.183.1.1 Equations 517
10.183.1.2 Rows 517
10.183.2 Constructor & Destructor Documentation 517
10.183.2.1 EquationRegistry() 517
10.183.3 Member Function Documentation 517
10.183.3.1 for_each() 517
10.183.3.2 get() 518
10.183.4 Member Data Documentation 518
10.183.4.1 m_equations 518
10.184 serif::operators::EquationRegistry< EquationTs > Class Template Reference 518
10.184.1 Member Typedef Documentation 518
10.184.1.1 Equations 518
10.184.1.2 Rows 519
10.184.2 Constructor & Destructor Documentation 519
10.184.2.1 EquationRegistry() 519
10.184.3 Member Function Documentation 519
10.184.3.1 for_each() 519
10.184.3.2 get() 519
10.184.4 Member Data Documentation 519
10.184.4.1 m_equations 519
10.185 serif::equations::EquationScale< EquationT > Class Template Reference 520
10.185.1 Detailed Description 521
10.185.2 Member Typedef Documentation 521
10.185.2.1 Test 521
10.185.3 Constructor & Destructor Documentation 521
10.185.3.1 EquationScale() 521
10.185.4 Member Function Documentation 521
10.185.4.1 coefficient() 521
10.185.4.2 contributions() 521
10.185.5 Member Data Documentation 521
10.185.5.1 m_coefficient 521
10.185.5.2 m_equation 522
10.185.5.3 name 522
10.186 serif::equations::EquationSum< FirstT, RestTs > Class Template Reference 522

10.186.1 Detailed Description . . . 523
10.186.2 Member Typedef Documentation . . . 523
10.186.2.1 Test . . . 523
10.186.3 Constructor & Destructor Documentation . . . 523
10.186.3.1 EquationSum() . . . 523
10.186.4 Member Function Documentation . . . 523
10.186.4.1 contributions() . . . 523
10.186.5 Member Data Documentation . . . 524
10.186.5.1 m_equations . . . 524
10.186.5.2 name . . . 524
10.187 serif::mapping::compactification::ExteriorDomainMap Class Reference . . . 524
10.187.1 Constructor & Destructor Documentation . . . 525
10.187.1.1 ~ExteriorDomainMap() . . . 525
10.187.2 Member Function Documentation . . . 525
10.187.2.1 clone() . . . 525
10.187.2.2 evaluate() . . . 525
10.187.2.3 evaluate_variation() . . . 525
10.187.2.4 get_name() . . . 525
10.187.2.5 is_affine_in_displacement() . . . 526
10.188 serif::mapping::compactification::ExteriorMapDirection Struct Reference . . . 526
10.188.1 Member Data Documentation . . . 526
10.188.1.1 displaced_position_variation . . . 526
10.188.1.2 displacement_jacobian_variation . . . 526
10.189 serif::mapping::compactification::ExteriorMapInput Struct Reference . . . 526
10.189.1 Member Data Documentation . . . 527
10.189.1.1 compactification_coordinate . . . 527
10.189.1.2 compactification_coordinate_gradient . . . 527
10.189.1.3 displaced_position . . . 527
10.189.1.4 displacement_jacobian . . . 527
10.189.1.5 reference_position . . . 527
10.190 serif::mapping::compactification::ExteriorMapResult Struct Reference . . . 527
10.190.1 Member Data Documentation . . . 528
10.190.1.1 mapping_jacobian . . . 528
10.190.1.2 physical_position . . . 528
10.191 serif::mapping::compactification::ExteriorMapVariation Struct Reference . . . 528
10.191.1 Member Data Documentation . . . 528
10.191.1.1 mapping_jacobian_variation . . . 528
10.191.1.2 physical_position_variation . . . 528
10.192 serif::discretization::domain::mesh::FaceElements Struct Reference . . . 529
10.192.1 Member Data Documentation . . . 529

10.192.1.1 firstElementID . . . 529
10.192.1.2 secondElementID . . . 529
10.193 serif::mapping::FaceMappingContext Struct Reference . . . 529
10.193.1 Member Data Documentation . . . 530
10.193.1.1 mapping . . . 530
10.193.1.2 physical_surface_weight . . . 530
10.193.1.3 quadrature . . . 530
10.193.1.4 reference_normal . . . 530
10.193.1.5 reference_surface_weight . . . 530
10.194 serif::mapping::FaceMappingVariation Struct Reference . . . 531
10.194.1 Member Data Documentation . . . 531
10.194.1.1 mapping . . . 531
10.194.1.2 normal_flux_scale_variation . . . 531
10.194.1.3 physical_normal_variation . . . 531
10.194.1.4 physical_surface_weight_variation . . . 532
10.195 serif::mapping::FaceQuadratureContext Struct Reference . . . 532
10.195.1 Member Data Documentation . . . 532
10.195.1.1 ds . . . 532
10.195.1.2 normal . . . 532
10.195.1.3 v_dot_n_scale . . . 532
10.196 serif::discretization::blocks::field Struct Reference . . . 533
10.196.1 Detailed Description . . . 533
10.197 serif::discretization::blocks::FieldDescriptor Struct Reference . . . 534
10.197.1 Detailed Description . . . 535
10.197.2 Friends And Related Symbol Documentation . . . 535
10.197.2.1 operator== . . . 535
10.197.3 Member Data Documentation . . . 535
10.197.3.1 name . . . 535
10.198 serif::utils::misc::concepts::Filter< List, Predicate > Struct Template Reference . . . 535
10.199 serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate > Struct Template Reference . . . 536
10.199.1 Member Typedef Documentation . . . 536
10.199.1.1 FilteredTail . . . 536
10.199.1.2 Type . . . 536
10.200 serif::utils::misc::concepts::Filter< types::TypeList<>, Predicate > Struct Template Reference 536
10.200.1 Member Typedef Documentation . . . 537
10.200.1.1 Type . . . 537
10.201 serif::geometry::deformation::FixedInfinityRadialExtension Struct Reference . . . 537
10.201.1 Detailed Description . . . 537
10.201.2 Member Function Documentation . . . 537

10.201.2.1 operator==() . . . 537
10.202 serif::dimensions::Force Struct Reference . . . 538
10.202.1 Detailed Description . . . 539
10.202.2 Member Data Documentation . . . 539
10.202.2.1 identifier . . . 539
10.203 serif::discretization::forms::force< TrialT, TestT > Struct Template Reference . . . 539
10.203.1 Detailed Description . . . 540
10.203.2 Member Typedef Documentation . . . 541
10.203.2.1 operand_types . . . 541
10.203.2.2 test_type . . . 541
10.203.2.3 trial_type . . . 541
10.203.3 Member Data Documentation . . . 541
10.203.3.1 name . . . 541
10.204 serif::discretization::forms::form Struct Reference . . . 542
10.204.1 Detailed Description . . . 542
10.205 std::formatter< serif::discretization::blocks::FieldDescriptor > Struct Reference . . . 543
10.205.1 Detailed Description . . . 544
10.205.2 Member Function Documentation . . . 544
10.205.2.1 format() . . . 544
10.206 std::formatter< serif::discretization::blocks::TermDescriptor > Struct Reference . . . 544
10.206.1 Detailed Description . . . 545
10.206.2 Member Function Documentation . . . 545
10.206.2.1 format() . . . 545
10.207 std::formatter< serif::discretization::domain::DomainOrSet > Struct Reference . . . 546
10.207.1 Member Function Documentation . . . 547
10.207.1.1 format() . . . 547
10.208 std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult > Struct Reference . . . 547
10.208.1 Member Function Documentation . . . 547
10.208.1.1 format() . . . 547
10.208.1.2 parse() . . . 547
10.208.2 Member Data Documentation . . . 548
10.208.2.1 string_formatter . . . 548
10.209 std::formatter< serif::discretization::forms::FormDescriptor > Struct Reference . . . 548
10.209.1 Detailed Description . . . 549
10.209.2 Member Function Documentation . . . 549
10.209.2.1 format() . . . 549
10.210 std::formatter< serif::discretization::functional_space::Continuity > Struct Reference . . . 550
10.210.1 Detailed Description . . . 551
10.210.2 Member Function Documentation . . . 551

10.210.2.1 format() . . . 551
10.211 std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > > Struct Template Reference . . . 551
10.211.1 Detailed Description . . . 552
10.211.2 Member Function Documentation . . . 552
10.211.2.1 format() . . . 552
10.212 std::formatter< serif::discretization::functional_space::Family > Struct Reference . . . 553
10.212.1 Detailed Description . . . 554
10.212.2 Member Function Documentation . . . 554
10.212.2.1 format() . . . 554
10.213 std::formatter< serif::discretization::functional_space::Mapping > Struct Reference . . . 554
10.213.1 Detailed Description . . . 555
10.213.2 Member Function Documentation . . . 555
10.213.2.1 format() . . . 555
10.214 std::formatter< serif::discretization::functional_space::Rank< R > > Struct Template Reference . . . 556
10.214.1 Detailed Description . . . 557
10.214.2 Member Function Documentation . . . 557
10.214.2.1 format() . . . 557
10.215 std::formatter< serif::discretization::functional_space::SpaceDescriptor > Struct Reference . . . 557
10.215.1 Detailed Description . . . 558
10.215.2 Member Function Documentation . . . 558
10.215.2.1 format() . . . 558
10.216 std::formatter< serif::discretization::functional_space::Trace > Struct Reference . . . 559
10.216.1 Detailed Description . . . 560
10.216.2 Member Function Documentation . . . 560
10.216.2.1 format() . . . 560
10.217 std::formatter< serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext > Struct Reference . . . 560
10.217.1 Member Function Documentation . . . 561
10.217.1.1 format() . . . 561
10.218 std::formatter< serif::discretization::quadrature::FormRule > Struct Reference . . . 561
10.218.1 Member Function Documentation . . . 562
10.218.1.1 format() . . . 562
10.219 std::formatter< serif::discretization::quadrature::MappingKind > Struct Reference . . . 563
10.219.1 Member Function Documentation . . . 564
10.219.1.1 format() . . . 564
10.220 std::formatter< serif::discretization::quadrature::Mode > Struct Reference . . . 564
10.220.1 Member Function Documentation . . . 565
10.220.1.1 format() . . . 565
10.221 std::formatter< serif::discretization::quadrature::QuadratureRole > Struct Reference . . . 565

10.221.1 Member Function Documentation . . . 566
10.221.1.1 format() . . . 566
10.222 std::formatter< serif::discretization::quadrature::Query > Struct Reference . . . 567
10.222.1 Member Function Documentation . . . 568
10.222.1.1 format() . . . 568
10.223 std::formatter< serif::discretization::quadrature::RuleControl > Struct Reference . . . 568
10.223.1 Member Function Documentation . . . 569
10.223.1.1 format() . . . 569
10.224 std::formatter< serif::discretization::quadrature::RuleSet > Struct Reference . . . 569
10.224.1 Member Function Documentation . . . 570
10.224.1.1 format() . . . 570
10.225 serif::discretization::forms::FormDescriptor Struct Reference . . . 571
10.225.1 Detailed Description . . . 572
10.225.2 Friends And Related Symbol Documentation . . . 572
10.225.2.1 operator== . . . 572
10.225.3 Member Data Documentation . . . 572
10.225.3.1 name . . . 572
10.225.3.2 test . . . 573
10.225.3.3 trial . . . 573
10.226 serif::discretization::quadrature::FormRule Struct Reference . . . 573
10.226.1 Member Data Documentation . . . 573
10.226.1.1 control . . . 573
10.226.1.2 form . . . 573
10.227 serif::discretization::forms::FormTraits< FormT > Struct Template Reference . . . 574
10.227.1 Detailed Description . . . 574
10.227.2 Member Data Documentation . . . 574
10.227.2.1 test_order_reduction . . . 574
10.227.2.2 trial_order_reduction . . . 574
10.228 serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > > Struct Template Reference . . . 575
10.228.1 Detailed Description . . . 575
10.228.2 Member Data Documentation . . . 575
10.228.2.1 test_order_reduction [1/2] . . . 575
10.228.2.2 test_order_reduction [2/2] . . . 575
10.228.2.3 trial_order_reduction [1/2] . . . 575
10.228.2.4 trial_order_reduction [2/2] . . . 576
10.229 serif::discretization::forms::FormTraits< divergence< TrialT, TestT > > Struct Template Reference . . . 576
10.229.1 Detailed Description . . . 576
10.229.2 Member Data Documentation . . . 576

10.229.2.1 test_order_reduction [1/2] . . . 576
10.229.2.2 test_order_reduction [2/2] . . . 577
10.229.2.3 trial_order_reduction [1/2] . . . 577
10.229.2.4 trial_order_reduction [2/2] . . . 577
10.230 serif::discretization::forms::FormTraits< force< TrialT, TestT > > Struct Template Reference 577
10.230.1 Detailed Description . . . 577
10.230.2 Member Data Documentation . . . 578
10.230.2.1 test_order_reduction [1/2] . . . 578
10.230.2.2 test_order_reduction [2/2] . . . 578
10.230.2.3 trial_order_reduction [1/2] . . . 578
10.230.2.4 trial_order_reduction [2/2] . . . 578
10.231 serif::discretization::forms::FormTraits< mass< TrialT, TestT > > Struct Template Reference 578
10.231.1 Detailed Description . . . 578
10.231.2 Member Data Documentation . . . 579
10.231.2.1 test_order_reduction [1/2] . . . 579
10.231.2.2 test_order_reduction [2/2] . . . 579
10.231.2.3 trial_order_reduction [1/2] . . . 579
10.231.2.4 trial_order_reduction [2/2] . . . 579
10.232 serif::discretization::forms::FormTraits< projection< TrialT, TestT > > Struct Template Reference . . . 579
10.232.1 Detailed Description . . . 580
10.232.2 Member Data Documentation . . . 581
10.232.2.1 test_order_reduction [1/2] . . . 581
10.232.2.2 test_order_reduction [2/2] . . . 581
10.232.2.3 trial_order_reduction [1/2] . . . 581
10.232.2.4 trial_order_reduction [2/2] . . . 581
10.233 serif::discretization::forms::FormTraits< source< TrialT, TestT > > Struct Template Reference 581
10.233.1 Detailed Description . . . 581
10.233.2 Member Data Documentation . . . 582
10.233.2.1 test_order_reduction [1/2] . . . 582
10.233.2.2 test_order_reduction [2/2] . . . 582
10.233.2.3 trial_order_reduction [1/2] . . . 582
10.233.2.4 trial_order_reduction [2/2] . . . 582
10.234 serif::dimensions::Frequency Struct Reference . . . 582
10.234.1 Detailed Description . . . 583
10.234.2 Member Data Documentation . . . 584
10.234.2.1 identifier . . . 584
10.235 serif::discretization::domain::relation::FullyConnected< A > Struct Template Reference . . . 584
10.235.1 Member Typedef Documentation . . . 585
10.235.1.1 domain_type . . . 585

10.235.2 Member Data Documentation . . . 585
10.235.2.1 name . . . 585
10.236 serif::discretization::domain::schema::validation::FullyConnected< A > Struct Template Reference . . . 586
10.236.1 Member Typedef Documentation . . . 587
10.236.1.1 domain_type . . . 587
10.236.2 Member Data Documentation . . . 587
10.236.2.1 name . . . 587
10.237 serif::discretization::blocks::generated_residual_block< GeneratedResidual > Struct Template Reference . . . 587
10.237.1 Detailed Description . . . 588
10.237.2 Member Typedef Documentation . . . 589
10.237.2.1 GeneratedType . . . 589
10.237.3 Member Data Documentation . . . 589
10.237.3.1 static_block_size . . . 589
10.238 serif::discretization::blocks::generated_value_block< GeneratedValue > Struct Template Reference . . . 589
10.238.1 Detailed Description . . . 590
10.238.2 Member Typedef Documentation . . . 591
10.238.2.1 GeneratedType . . . 591
10.238.3 Member Data Documentation . . . 591
10.238.3.1 static_block_size . . . 591
10.239 serif::operators::backend::mfem::volume_keys::Geometry< Contribution > Struct Template Reference . . . 591
10.240 serif::geometry::deformation::backend::mfem::GeometrySampleRule Struct Reference . . . 591
10.240.1 Detailed Description . . . 592
10.240.2 Member Data Documentation . . . 592
10.240.2.1 element . . . 592
10.240.2.2 integration_rule . . . 592
10.241 serif::geometry::deformation::GeometryStepLimit Struct Reference . . . 592
10.241.1 Detailed Description . . . 592
10.241.2 Member Data Documentation . . . 593
10.241.2.1 boundary_step . . . 593
10.241.2.2 limited_by_geometry . . . 593
10.241.2.3 limiting_element . . . 593
10.241.2.4 limiting_point . . . 593
10.241.2.5 limiting_point_determinant_at_step . . . 593
10.241.2.6 limiting_rule . . . 593
10.241.2.7 minimum_determinant_at_accepted . . . 593
10.241.2.8 minimum_determinant_at_maximum . . . 593
10.241.2.9 sampled_point_count . . . 593

10.241.2.10 step . . . 594
10.242 serif::geometry::deformation::GeometryStepOptions Struct Reference . . . 594
10.242.1 Detailed Description . . . 594
10.242.2 Member Data Documentation . . . 594
10.242.2.1 determinant_floor . . . 594
10.242.2.2 fraction_to_boundary . . . 594
10.242.2.3 maximum_step . . . 594
10.243 serif::discretization::blocks::global_scalar Struct Reference . . . 595
10.243.1 Detailed Description . . . 595
10.244 serif::discretization::forms::Operation::Gradient Struct Reference . . . 596
10.244.1 Detailed Description . . . 596
10.245 serif::discretization::blocks::gravity::gradient Struct Reference . . . 596
10.245.1 Detailed Description . . . 598
10.245.2 Member Typedef Documentation . . . 598
10.245.2.1 FunctionalSpace . . . 598
10.245.2.2 Quantity . . . 598
10.245.2.3 Support . . . 598
10.245.3 Member Data Documentation . . . 598
10.245.3.1 name . . . 598
10.245.3.2 symbol . . . 599
10.246 serif::equations::gravity_contributions::GradientDivergence Struct Reference . . . 599
10.246.1 Member Typedef Documentation . . . 600
10.246.1.1 Operands . . . 600
10.246.2 Member Function Documentation . . . 600
10.246.2.1 additional_order() . . . 600
10.246.2.2 evaluate() . . . 600
10.247 serif::operators::GradientRequest Struct Reference . . . 601
10.247.1 Member Typedef Documentation . . . 601
10.247.1.1 Operands . . . 601
10.247.1.2 Test . . . 602
10.247.2 Member Function Documentation . . . 602
10.247.2.1 additional_order() . . . 602
10.247.2.2 evaluate() . . . 602
10.247.3 Member Data Documentation . . . 602
10.247.3.1 name . . . 602
10.248 serif::equations::gravity_contributions::GradientValue Struct Reference . . . 603
10.248.1 Member Typedef Documentation . . . 604
10.248.1.1 Operands . . . 604
10.248.2 Member Function Documentation . . . 604
10.248.2.1 additional_order() . . . 604

10.248.2.2 evaluate() . . . 604
10.249 serif::utils::dependencies::detail::Graph< Computations > Struct Template Reference . . . 605
10.249.1 Member Typedef Documentation . . . 606
10.249.1.1 Mask . . . 606
10.249.1.2 Matrix . . . 606
10.249.1.3 Order . . . 606
10.249.2 Member Function Documentation . . . 606
10.249.2.1 acyclic() . . . 606
10.249.2.2 make_order() . . . 606
10.249.2.3 prerequisites() . . . 606
10.249.2.4 required() . . . 607
10.249.3 Member Data Documentation . . . 607
10.249.3.1 edges . . . 607
10.249.3.2 order . . . 607
10.249.3.3 size . . . 607
10.250 serif::dimensions::GravitationalPotential Struct Reference . . . 607
10.250.1 Detailed Description . . . 608
10.250.2 Member Data Documentation . . . 609
10.250.2.1 identifier . . . 609
10.251 serif::discretization::blocks::gravity Struct Reference . . . 609
10.251.1 Detailed Description . . . 610
10.251.2 Member Typedef Documentation . . . 610
10.251.2.1 Terms . . . 610
10.251.3 Member Data Documentation . . . 611
10.251.3.1 gradient_term . . . 611
10.251.3.2 name . . . 611
10.251.3.3 potential_term . . . 611
10.252 serif::equations::GravityForce Struct Reference . . . 611
10.252.1 Detailed Description . . . 612
10.252.2 Member Typedef Documentation . . . 612
10.252.2.1 Density . . . 612
10.252.2.2 DensityValue . . . 612
10.252.2.3 Gradient . . . 613
10.252.2.4 GradientValue . . . 613
10.252.2.5 Operands . . . 613
10.252.2.6 Result . . . 613
10.252.3 Member Function Documentation . . . 613
10.252.3.1 additional_order() . . . 613
10.252.3.2 evaluate() . . . 613
10.252.4 Member Data Documentation . . . 613

10.252.4.1 name . . . 613
10.253 serif::equations::GravityGradientRelation Struct Reference . . . 614
10.253.1 Member Typedef Documentation . . . 614
10.253.1.1 Test . . . 614
10.253.2 Member Function Documentation . . . 615
10.253.2.1 contributions() . . . 615
10.253.3 Member Data Documentation . . . 615
10.253.3.1 name . . . 615
10.254 serif::mapping::GridFunctionMappingEvaluator Class Reference . . . 615
10.254.1 Constructor & Destructor Documentation . . . 616
10.254.1.1 GridFunctionMappingEvaluator() . . . 616
10.254.2 Member Function Documentation . . . 616
10.254.2.1 evaluate_face() . . . 616
10.254.2.2 evaluate_point() . . . 617
10.254.2.3 evaluate_volume() . . . 617
10.254.2.4 get_face_quadrature_context() . . . 617
10.254.2.5 get_physical_point() . . . 617
10.254.2.6 get_quadrature_context() . . . 617
10.254.2.7 invalidate_cache() . . . 617
10.254.2.8 invalidate_for_changed_spaces() . . . 617
10.254.2.9 load_element() . . . 618
10.254.2.10 refresh() . . . 618
10.254.2.11 validate_field_bindings() . . . 618
10.254.3 Member Data Documentation . . . 618
10.254.3.1 m_cached_element_id . . . 618
10.254.3.2 m_compactification_coordinate . . . 618
10.254.3.3 m_compactification_data . . . 618
10.254.3.4 m_compactification_dofs . . . 618
10.254.3.5 m_compactification_space . . . 618
10.254.3.6 m_compactification_space_sequence . . . 618
10.254.3.7 m_displacement . . . 619
10.254.3.8 m_displacement_data . . . 619
10.254.3.9 m_displacement_dofs . . . 619
10.254.3.10 m_displacement_space . . . 619
10.254.3.11 m_displacement_space_sequence . . . 619
10.254.3.12 m_element_compactification . . . 619
10.254.3.13 m_element_displacement . . . 619
10.254.3.14 m_mapper . . . 619
10.254.3.15 m_workspace . . . 619
10.255 serif::utils::dependencies::detail::HasBinding< BindingsT, KeyT > Struct Template Reference 620

10.256 serif::utils::dependencies::HasBinding< BindingsT, KeyT > Struct Template Reference . . . 620
10.257 serif::utils::dependencies::detail::HasBinding< Bindings< BindingTs... >, KeyT > Struct Template Reference . . . 621
10.257.1 Member Data Documentation . . . 621
10.257.1.1 value . . . 621
10.258 serif::eos::runtime::HasRuntimeQuantityIdentifiers< RelationT > Struct Template Reference 621
10.259 serif::eos::runtime::HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > > Struct Template Reference . . . 622
10.260 serif::operators::HasScalarVolumeOperands< EquationT, OperandsT > Struct Template Reference . . . 623
10.261 serif::operators::HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > > Struct Template Reference . . . 624
10.262 serif::eos::models::ImplementsPartialDerivative_t< EOS, RelationT, InputQuantity > Struct Template Reference . . . 625
10.263 serif::eos::models::ImplementsRelation_t< EOS, RelationT > Struct Template Reference . 626
10.264 serif::eos::models::ImplementsRelationCatalog_t< EOS, Output, Inputs > Struct Template Reference . . . 627
10.265 serif::eos::models::ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > > Struct Template Reference . . . 628
10.266 serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT > Class Template Reference . . . 629
10.266.1 Detailed Description . . . 630
10.266.2 Member Typedef Documentation . . . 630
10.266.2.1 Function . . . 630
10.266.2.2 Inputs . . . 630
10.266.2.3 Key . . . 630
10.266.2.4 Update . . . 630
10.266.3 Constructor & Destructor Documentation . . . 630
10.266.3.1 IncrementalComputation() . . . 630
10.266.4 Member Function Documentation . . . 631
10.266.4.1 evaluate() . . . 631
10.266.4.2 evaluate_update() . . . 631
10.266.5 Member Data Documentation . . . 631
10.266.5.1 m_function . . . 631
10.267 serif::discretization::domain::InfinitySurfaceBoundary Struct Reference . . . 631
10.267.1 Member Data Documentation . . . 632
10.267.1.1 name . . . 632
10.268 serif::utils::dependencies::keys::Input Struct Reference . . . 633
10.268.1 Detailed Description . . . 633
10.269 serif::discretization::domain::relation::Inscribed< A, B > Struct Template Reference . . . 633
10.269.1 Member Typedef Documentation . . . 634
10.269.1.1 inner_type . . . 634

10.269.1.2 outer_type . . . 634
10.269.2 Member Data Documentation . . . 635
10.269.2.1 name . . . 635
10.270 serif::discretization::domain::schema::validation::Inscribed< A, B > Struct Template Reference 635
10.270.1 Member Typedef Documentation . . . 636
10.270.1.1 inner_type . . . 636
10.270.1.2 outer_type . . . 636
10.270.2 Member Data Documentation . . . 637
10.270.2.1 name . . . 637
10.271 serif::discretization::domain::relation::validation::RelationValidationResult::Inscribed↩ Diagnostics Struct Reference . . . 637
10.271.1 Member Data Documentation . . . 637
10.271.1.1 adjacentDomainID . . . 637
10.271.1.2 adjacentElementID . . . 637
10.271.1.3 faceID . . . 637
10.271.1.4 innerElementID . . . 637
10.272 serif::discretization::domain::schema::validation::RelationValidationResult::Inscribed↩ Diagnostics Struct Reference . . . 638
10.272.1 Member Data Documentation . . . 638
10.272.1.1 adjacentDomainID . . . 638
10.272.1.2 adjacentElementID . . . 638
10.272.1.3 faceID . . . 638
10.272.1.4 innerElementID . . . 638
10.273 serif::geometry::deformation::keys::InteriorModel Struct Reference . . . 638
10.274 serif::dimensions::InternalEnergy Struct Reference . . . 639
10.274.1 Detailed Description . . . 640
10.274.2 Member Data Documentation . . . 640
10.274.2.1 identifier . . . 640
10.275 serif::utils::dependencies::detail::Invocation< ComputationT, ArgumentsT, typename > Struct Template Reference . . . 640
10.275.1 Member Data Documentation . . . 640
10.275.1.1 valid . . . 640
10.276 serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< decltype(std::declval< const ComputationT & >().evaluate(std::declval< const Arguments & >()...))> > Struct Template Reference . . . 641
10.276.1 Member Data Documentation . . . 642
10.276.1.1 valid . . . 642
10.277 serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, typename > Struct Template Reference . . . 642
10.278 serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std::remove_cvref_t< EOSCandidate >::Relations > > Struct Template Reference . . . 643
10.279 serif::discretization::blocks::IsGlobalScalarTerm< T > Struct Template Reference . . . 644

10.279.1 Detailed Description . . . 645
10.280 serif::geometry::surface::IsobaricSurface Class Reference . . . 645
10.280.1 Member Typedef Documentation . . . 646
10.280.1.1 PhysicalQuantity . . . 646
10.280.1.2 TargetValue . . . 647
10.280.2 Constructor & Destructor Documentation . . . 647
10.280.2.1 IsobaricSurface() [1/2] . . . 647
10.280.2.2 IsobaricSurface() [2/2] . . . 647
10.280.3 Member Function Documentation . . . 647
10.280.3.1 continuation_identity() . . . 647
10.280.3.2 descriptor() . . . 647
10.280.3.3 target_pressure() . . . 647
10.280.4 Member Data Documentation . . . 647
10.280.4.1 m_targetPressure . . . 647
10.281 serif::discretization::blocks::IsPhysicalSpatialTerm< T > Struct Template Reference . . . 648
10.281.1 Detailed Description . . . 649
10.282 serif::dimensions::IsQuantityValue< Candidate > Struct Template Reference . . . 650
10.282.1 Detailed Description . . . 651
10.283 serif::dimensions::IsQuantityValue< QuantityValue< Quantity > > Struct Template Reference 651
10.284 serif::discretization::blocks::IsSpatialTerm< T > Struct Template Reference . . . 652
10.284.1 Detailed Description . . . 653
10.285 serif::eos::relations::IsThermodynamicRelation< RelationCandidate > Struct Template Reference . . . 653
10.286 serif::eos::relations::IsThermodynamicRelation< Relation< Output, Inputs... > > Struct Template Reference . . . 654
10.287 serif::eos::relations::IsValidRelationCatalog< CandidateCatalog > Struct Template Reference 655
10.288 serif::eos::relations::IsValidRelationCatalog< RelationCatalog< Relations... > > Struct Template Reference . . . 656
10.289 serif::operators::backend::mfem::JacobianOperator< CacheT > Class Template Reference . . . 657
10.289.1 Detailed Description . . . 659
10.289.2 Constructor & Destructor Documentation . . . 659
10.289.2.1 JacobianOperator() [1/3] . . . 659
10.289.2.2 JacobianOperator() [2/3] . . . 659
10.289.2.3 JacobianOperator() [3/3] . . . 659
10.289.3 Member Function Documentation . . . 659
10.289.3.1 checked_data() . . . 659
10.289.3.2 Mult() . . . 660
10.289.3.3 MultTranspose() . . . 660
10.289.4 Member Data Documentation . . . 660
10.289.4.1 m_cache . . . 660
10.289.4.2 m_workspace . . . 660

10.290 serif::operators::detail::Join< Lists > Struct Template Reference 660
10.290.1 Member Typedef Documentation 661
10.290.1.1 type 661
10.291 serif::mapping::compactification::KelvinCompactification Class Reference 661
10.291.1 Constructor & Destructor Documentation 662
10.291.1.1 KelvinCompactification() 662
10.291.2 Member Function Documentation 663
10.291.2.1 clone() 663
10.291.2.2 compute_radial_factors() 663
10.291.2.3 evaluate() 663
10.291.2.4 evaluate_jacobian_variation_factors() 663
10.291.2.5 evaluate_variation() 663
10.291.2.6 get_coordinate_tolerance() 663
10.291.2.7 get_name() 664
10.291.2.8 get_reference_infinity_radius() 664
10.291.2.9 get_reference_stellar_radius() 664
10.291.2.10 is_affine_in_displacement() 664
10.291.3 Member Data Documentation 664
10.291.3.1 m_name 664
10.291.3.2 m_options 664
10.292 serif::utils::user_options::KelvinCompactificationOptions Struct Reference 665
10.292.1 Member Data Documentation 665
10.292.1.1 coordinate_tolerance 665
10.292.1.2 r_inf_ref 665
10.292.1.3 r_star_ref 665
10.293 serif::utils::dependencies::KeyList< KeyTs > Struct Template Reference 665
10.293.1 Detailed Description 666
10.293.2 Member Typedef Documentation 666
10.293.2.1 Keys 666
10.294 serif::utils::dependencies::KeyListBindings< BindingsT, KeysT > Struct Template Reference 666
10.295 serif::utils::dependencies::KeyListBindings< BindingsT, KeyList< KeyTs... > > Struct Template Reference 666
10.296 serif::utils::dependencies::detail::KeyListSize< KeysT > Struct Template Reference 667
10.297 serif::utils::dependencies::detail::KeyListSize< KeyList< Keys... > > Struct Template Reference 667
10.298 serif::dimensions::KinematicViscosity Struct Reference 668
10.298.1 Detailed Description 669
10.298.2 Member Data Documentation 670
10.298.2.1 identifier 670
10.299 serif::dimensions::Length Struct Reference 670

10.299.1 Detailed Description . . . 671
10.299.2 Member Data Documentation . . . 671
10.299.2.1 identifier . . . 671
10.300 serif::geometry::deformation::keys::Linearization Struct Reference . . . 671
10.301 serif::geometry::surface::keys::Linearization Struct Reference . . . 672
10.302 serif::operators::backend::mfem::nodal_keys::Linearization Struct Reference . . . 672
10.303 serif::operators::backend::mfem::volume_keys::Linearization Struct Reference . . . 672
10.304 serif::geometry::deformation::backend::mfem::detail::LocatedBoundaryPoint Struct Reference 672
10.304.1 Detailed Description . . . 672
10.304.2 Member Data Documentation . . . 673
10.304.2.1 boundary_element . . . 673
10.304.2.2 point . . . 673
10.305 serif::dimensions::Luminosity Struct Reference . . . 673
10.305.1 Detailed Description . . . 674
10.305.2 Member Data Documentation . . . 674
10.305.2.1 identifier . . . 674
10.306 serif::dimensions::MagneticFluxDensity Struct Reference . . . 675
10.306.1 Detailed Description . . . 676
10.306.2 Member Data Documentation . . . 676
10.306.2.1 identifier . . . 676
10.307 serif::operators::backend::mfem::MappingCoefficients Struct Reference . . . 676
10.307.1 Member Data Documentation . . . 676
10.307.1.1 binding . . . 676
10.307.1.2 values . . . 676
10.308 serif::mapping::MappingPointContext Struct Reference . . . 676
10.308.1 Member Data Documentation . . . 677
10.308.1.1 compactified . . . 677
10.308.1.2 displaced_position . . . 677
10.308.1.3 displacement_jacobian . . . 677
10.308.1.4 inverse_mapping_jacobian . . . 677
10.308.1.5 mapping_determinate . . . 677
10.308.1.6 mapping_jacobian . . . 677
10.308.1.7 physical_position . . . 677
10.308.1.8 reference_position . . . 678
10.309 serif::mapping::MappingPointVariation Struct Reference . . . 678
10.309.1 Member Data Documentation . . . 678
10.309.1.1 displacement_jacobian_variation . . . 678
10.309.1.2 displacement_variation . . . 678
10.309.1.3 inverse_mapping_jacobian_variation . . . 678
10.309.1.4 mapping_determinate_variation . . . 678

10.309.1.5 mapping_jacobian_variation . . . 679
10.309.1.6 physical_position_variation . . . 679
10.310 serif::dimensions::Mass Struct Reference . . . 679
10.310.1 Detailed Description . . . 680
10.310.2 Member Data Documentation . . . 680
10.310.2.1 identifier . . . 680
10.311 serif::discretization::blocks::density::mass Struct Reference . . . 681
10.311.1 Detailed Description . . . 683
10.311.2 Member Typedef Documentation . . . 683
10.311.2.1 FunctionalSpace . . . 683
10.311.2.2 Quantity . . . 683
10.311.2.3 Support . . . 683
10.311.3 Member Data Documentation . . . 683
10.311.3.1 name . . . 683
10.311.3.2 symbol . . . 684
10.312 serif::discretization::forms::mass< TrialT, TestT > Struct Template Reference . . . 684
10.312.1 Detailed Description . . . 685
10.312.2 Member Typedef Documentation . . . 687
10.312.2.1 operand_types . . . 687
10.312.2.2 test_type . . . 687
10.312.2.3 trial_type . . . 687
10.312.3 Member Data Documentation . . . 687
10.312.3.1 name . . . 687
10.313 serif::discretization::blocks::barotropic_constant::mass_normalization Struct Reference . . . 688
10.313.1 Detailed Description . . . 689
10.313.2 Member Typedef Documentation . . . 689
10.313.2.1 Quantity . . . 689
10.313.2.2 Support . . . 689
10.313.3 Member Data Documentation . . . 689
10.313.3.1 name . . . 689
10.313.3.2 symbol . . . 689
10.314 serif::dimensions::MassDensity Struct Reference . . . 690
10.314.1 Detailed Description . . . 691
10.314.2 Member Data Documentation . . . 691
10.314.2.1 identifier . . . 691
10.315 serif::dimensions::MassFlowRate Struct Reference . . . 691
10.315.1 Detailed Description . . . 692
10.315.2 Member Data Documentation . . . 692
10.315.2.1 identifier . . . 692
10.316 serif::equations::MassNormalization Class Reference . . . 693

10.316.1 Detailed Description . . . 694
10.316.2 Member Typedef Documentation . . . 694
10.316.2.1 Density . . . 694
10.316.2.2 DensityValue . . . 694
10.316.2.3 Operands . . . 694
10.316.3 Constructor & Destructor Documentation . . . 694
10.316.3.1 MassNormalization() . . . 694
10.316.4 Member Function Documentation . . . 694
10.316.4.1 additional_order() . . . 694
10.316.4.2 evaluate() . . . 695
10.316.4.3 residual_offset() . . . 695
10.316.5 Member Data Documentation . . . 695
10.316.5.1 m_target_mass . . . 695
10.316.5.2 name . . . 695
10.317 serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, Catalog > Struct Template Reference . . . 695
10.318 serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Relations... > > Struct Template Reference . . . 695
10.318.1 Member Typedef Documentation . . . 696
10.318.1.1 Matches . . . 696
10.318.2 Member Data Documentation . . . 696
10.318.2.1 count . . . 696
10.319 serif::utils::dependencies::detail::MergeKeyLists< ResultT, Lists > Struct Template Reference 696
10.319.1 Member Typedef Documentation . . . 697
10.319.1.1 type . . . 697
10.320 serif::utils::dependencies::detail::MergeKeyLists< ResultT, KeyList< Keys... >, Rest... > Struct Template Reference . . . 697
10.320.1 Member Typedef Documentation . . . 698
10.320.1.1 type . . . 698
10.321 serif::operators::backend::mfem::MeshAndCoefficientBindings Struct Reference . . . 698
10.322 serif::discretization::domain::mesh::MeshTopology Class Reference . . . 698
10.322.1 Constructor & Destructor Documentation . . . 699
10.322.1.1 MeshTopology() . . . 699
10.322.2 Member Function Documentation . . . 699
10.322.2.1 boundary_element_boundary_id() . . . 699
10.322.2.2 boundary_element_count() . . . 699
10.322.2.3 boundary_element_face() . . . 700
10.322.2.4 boundary_elements_on_face() . . . 700
10.322.2.5 element_count() . . . 700
10.322.2.6 element_domain_id() . . . 700
10.322.2.7 element_neighbors() . . . 700

10.322.2.8 face_count() . . . 700
10.322.2.9 face_elements() . . . 700
10.322.3 Member Data Documentation . . . 700
10.322.3.1 m_boundaryElementBoundaryIDs . . . 700
10.322.3.2 m_boundaryElementFaceIDs . . . 701
10.322.3.3 m_boundaryElementsByFace . . . 701
10.322.3.4 m_elementDomainIDs . . . 701
10.322.3.5 m_elementNeighbors . . . 701
10.322.3.6 m_faceElements . . . 701
10.322.3.7 s_noElements . . . 701
10.323 serif::operators::backend::mfem::MetadataTracking Struct Reference . . . 702
10.323.1 Detailed Description . . . 703
10.323.2 Member Function Documentation . . . 703
10.323.2.1 identity() . . . 703
10.324 serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< OperandListT > Struct Template Reference . . . 703
10.325 serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< utils::types::TypeList< OperandTs... > > Struct Template Reference . . . 703
10.325.1 Member Data Documentation . . . 704
10.325.1.1 value . . . 704
10.326 serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext Struct Reference 704
10.326.1 Member Data Documentation . . . 705
10.326.1.1 domain . . . 705
10.326.1.2 dynamic_orders . . . 706
10.326.1.3 geometry . . . 706
10.326.1.4 geometry_weight_order . . . 706
10.326.1.5 mapping . . . 706
10.326.1.6 role . . . 706
10.327 serif::discretization::quadrature::backend::mfem::MFEMResolver Class Reference . . . 706
10.327.1 Constructor & Destructor Documentation . . . 707
10.327.1.1 MFEMResolver() . . . 707
10.327.2 Member Function Documentation . . . 707
10.327.2.1 configure() . . . 707
10.327.2.2 resolve() . . . 707
10.327.3 Member Data Documentation . . . 707
10.327.3.1 m_policy . . . 707
10.328 serif::discretization::quadrature::backend::mfem::MFEMRule Struct Reference . . . 708
10.328.1 Member Data Documentation . . . 708
10.328.1.1 integration_rule . . . 708
10.328.1.2 resolution . . . 708

10.329 serif::dimensions::MomentOfInertia Struct Reference . . . 709
10.329.1 Detailed Description . . . 710
10.329.2 Member Data Documentation . . . 710
10.329.2.1 identifier . . . 710
10.330 serif::dimensions::Momentum Struct Reference . . . 710
10.330.1 Detailed Description . . . 711
10.330.2 Member Data Documentation . . . 711
10.330.2.1 identifier . . . 711
10.331 serif::operators::NestedBadCollection Struct Reference . . . 712
10.331.1 Member Typedef Documentation . . . 712
10.331.1.1 Test . . . 712
10.331.2 Member Function Documentation . . . 713
10.331.2.1 contributions() . . . 713
10.331.3 Member Data Documentation . . . 713
10.331.3.1 name . . . 713
10.332 serif::operators::backend::mfem::NodalData< EquationT, SchemaT > Struct Template Reference . . . 713
10.332.1 Detailed Description . . . 713
10.332.2 Member Typedef Documentation . . . 714
10.332.2.1 Discretization . . . 714
10.332.3 Member Data Documentation . . . 714
10.332.3.1 derivative . . . 714
10.332.3.2 discretization . . . 714
10.332.3.3 residual . . . 714
10.333 serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT > Struct Template Reference . . . 714
10.333.1 Detailed Description . . . 715
10.333.2 Member Typedef Documentation . . . 715
10.333.2.1 Equation . . . 715
10.333.2.2 ResidualBinding . . . 715
10.333.2.3 Test . . . 715
10.333.2.4 Value . . . 716
10.333.3 Member Data Documentation . . . 716
10.333.3.1 components . . . 716
10.333.3.2 residual_binding . . . 716
10.333.3.3 selected_rows . . . 716
10.334 serif::geometry::deformation::NodalRadialSurface Struct Reference . . . 716
10.334.1 Detailed Description . . . 717
10.334.2 Member Function Documentation . . . 717
10.334.2.1 operator==() . . . 717

10.334.3 Member Data Documentation . . . 717
10.334.3.1 reference_center . . . 717
10.335 serif::operators::NodalRequest< TermT, OperationT, PositionIndependent > Struct Template Reference . . . 718
10.335.1 Member Typedef Documentation . . . 719
10.335.1.1 Operands . . . 719
10.335.1.2 Result . . . 719
10.335.2 Member Function Documentation . . . 719
10.335.2.1 evaluate() . . . 719
10.335.3 Member Data Documentation . . . 719
10.335.3.1 name . . . 719
10.335.3.2 position_independent . . . 719
10.336 serif::operators::backend::mfem::NodalValueTracking Struct Reference . . . 720
10.336.1 Detailed Description . . . 721
10.336.2 Member Function Documentation . . . 721
10.336.2.1 identity() . . . 721
10.337 serif::discretization::forms::Operation::NormalTrace Struct Reference . . . 721
10.337.1 Detailed Description . . . 722
10.338 serif::dimensions::NumberDensity Struct Reference . . . 722
10.338.1 Detailed Description . . . 723
10.338.2 Member Data Documentation . . . 723
10.338.2.1 identifier . . . 723
10.339 serif::operators::backend::mfem::NumericalWork Struct Reference . . . 723
10.339.1 Detailed Description . . . 724
10.339.2 Member Function Documentation . . . 724
10.339.2.1 did_work() . . . 724
10.339.3 Member Data Documentation . . . 724
10.339.3.1 geometry_points . . . 724
10.339.3.2 physics_points . . . 724
10.339.3.3 refreshed_scalars . . . 724
10.339.3.4 reused_elements . . . 724
10.339.3.5 sampled_operands . . . 725
10.340 serif::dimensions::Opacity Struct Reference . . . 725
10.340.1 Detailed Description . . . 726
10.340.2 Member Data Documentation . . . 726
10.340.2.1 identifier . . . 726
10.341 serif::discretization::forms::Operand< TermT, OperationT > Struct Template Reference . . . 727
10.341.1 Detailed Description . . . 727
10.341.2 Member Typedef Documentation . . . 727
10.341.2.1 operation_type . . . 727

10.341.2.2 term_type . . . 728
10.342 serif::operators::backend::mfem::OperandEvaluator< OperandT > Struct Template Reference 728
10.343 serif::operators::backend::mfem::OperandEvaluator< OperandT > Struct Template Reference 728
10.343.1 Detailed Description . . . 729
10.343.2 Member Typedef Documentation . . . 729
10.343.2.1 ReferenceValue . . . 729
10.343.2.2 Term . . . 730
10.343.2.3 Value . . . 730
10.343.3 Member Function Documentation . . . 730
10.343.3.1 evaluate() . . . 730
10.343.3.2 geometry_variation() . . . 730
10.343.3.3 gradient_trace() . . . 730
10.343.3.4 integrate() . . . 731
10.343.3.5 map_basis_point() . . . 731
10.343.3.6 physical_value() . . . 731
10.343.3.7 reference_value() . . . 731
10.343.3.8 sample_basis() [1/2] . . . 731
10.343.3.9 sample_basis() [2/2] . . . 732
10.343.4 Member Data Documentation . . . 732
10.343.4.1 components . . . 732
10.343.4.2 divergence . . . 732
10.343.4.3 geometry_dependent . . . 732
10.343.4.4 global_scalar . . . 732
10.343.4.5 h1_divergence . . . 732
10.343.4.6 hdiv . . . 733
10.343.4.7 reference_components . . . 733
10.343.4.8 vector_copies . . . 733
10.344 serif::discretization::forms::OperandResult< OperandT > Struct Template Reference . . . 733
10.344.1 Detailed Description . . . 733
10.345 serif::discretization::forms::OperandResult< OperandT > Struct Template Reference . . . 734
10.345.1 Detailed Description . . . 735
10.345.2 Member Typedef Documentation . . . 735
10.345.2.1 type . . . 735
10.346 serif::operators::backend::mfem::OperandSamples Struct Reference . . . 735
10.346.1 Detailed Description . . . 736
10.346.2 Member Data Documentation . . . 736
10.346.2.1 physical . . . 736
10.346.2.2 reference . . . 736
10.347 serif::equations::OperandValues< OperandsT, OutputT > Struct Template Reference . . . 736
10.348 serif::operators::OperandValues< OperandsT, OutputT > Struct Template Reference . . . 736

10.349 serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT > Struct Template Reference . . . 737
10.349.1 Member Typedef Documentation . . . 737
10.349.1.1 Entry . . . 737
10.349.1.2 Storage . . . 738
10.349.2 Member Function Documentation . . . 738
10.349.2.1 get() [1/2] . . . 738
10.349.2.2 get() [2/2] . . . 738
10.349.2.3 index() . . . 738
10.349.3 Member Data Documentation . . . 738
10.349.3.1 count . . . 738
10.349.3.2 values . . . 739
10.350 serif::discretization::forms::Operation Struct Reference . . . 739
10.350.1 Detailed Description . . . 739
10.351 serif::discretization::forms::OperationValue< Rank, OperationT > Struct Template Reference 740
10.351.1 Detailed Description . . . 740
10.352 serif::discretization::forms::OperationValue< 0, Operation::Gradient > Struct Reference . . 740
10.352.1 Detailed Description . . . 740
10.352.2 Member Typedef Documentation . . . 741
10.352.2.1 type . . . 741
10.353 serif::discretization::forms::OperationValue< 0, Operation::Value > Struct Reference . . . 741
10.353.1 Detailed Description . . . 741
10.353.2 Member Typedef Documentation . . . 741
10.353.2.1 type . . . 741
10.354 serif::discretization::forms::OperationValue< 1, Operation::Curl > Struct Reference . . . 741
10.354.1 Detailed Description . . . 742
10.354.2 Member Typedef Documentation . . . 742
10.354.2.1 type . . . 742
10.355 serif::discretization::forms::OperationValue< 1, Operation::Divergence > Struct Reference . 742
10.355.1 Detailed Description . . . 742
10.355.2 Member Typedef Documentation . . . 742
10.355.2.1 type . . . 742
10.356 serif::discretization::forms::OperationValue< 1, Operation::NormalTrace > Struct Reference 743
10.356.1 Detailed Description . . . 743
10.356.2 Member Typedef Documentation . . . 743
10.356.2.1 type . . . 743
10.357 serif::discretization::forms::OperationValue< 1, Operation::Value > Struct Reference . . . 743
10.357.1 Detailed Description . . . 743
10.357.2 Member Typedef Documentation . . . 744
10.357.2.1 type . . . 744

10.358 serif::utils::dependencies::keys::Output Struct Reference . . . 744
10.358.1 Detailed Description . . . 744
10.359 serif::utils::dependencies::OwnedComputation< KeyT, ComputationT > Class Template Reference . . . 744
10.359.1 Member Typedef Documentation . . . 745
10.359.1.1 Key . . . 745
10.359.1.2 Target . . . 745
10.359.2 Constructor & Destructor Documentation . . . 746
10.359.2.1 OwnedComputation() [1/3] . . . 746
10.359.2.2 OwnedComputation() [2/3] . . . 746
10.359.2.3 OwnedComputation() [3/3] . . . 746
10.359.3 Member Function Documentation . . . 746
10.359.3.1 checked_value() . . . 746
10.359.3.2 get() [1/4] . . . 746
10.359.3.3 get() [2/4] . . . 747
10.359.3.4 get() [3/4] . . . 747
10.359.3.5 get() [4/4] . . . 747
10.359.3.6 operator=() [1/2] . . . 747
10.359.3.7 operator=() [2/2] . . . 747
10.359.4 Member Data Documentation . . . 747
10.359.4.1 m_value . . . 747
10.360 serif::utils::dependencies::detail::OwnedValue< T > Struct Template Reference . . . 748
10.360.1 Member Typedef Documentation . . . 748
10.360.1.1 Target . . . 748
10.360.2 Member Function Documentation . . . 748
10.360.2.1 get() . . . 748
10.360.3 Member Data Documentation . . . 748
10.360.3.1 value . . . 748
10.361 serif::geometry::surface::IsobaricSurface::Parameters Struct Reference . . . 749
10.361.1 Member Data Documentation . . . 749
10.361.1.1 PSurf . . . 749
10.362 serif::utils::dependencies::keys::Parameters Struct Reference . . . 749
10.362.1 Detailed Description . . . 749
10.363 serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity > Class Template Reference . . . 749
10.363.1 Detailed Description . . . 751
10.363.2 Constructor & Destructor Documentation . . . 751
10.363.2.1 PartialDerivative() . . . 751
10.363.3 Member Function Documentation . . . 751
10.363.3.1 value() . . . 751

10.363.4 Friends And Related Symbol Documentation . . . 752
10.363.4.1 operator* [1/2] . . . 752
10.363.4.2 operator* [2/2] . . . 752
10.363.4.3 operator+ . . . 753
10.363.4.4 operator- . . . 753
10.363.4.5 operator/ . . . 753
10.363.4.6 operator<=> [1/3] . . . 754
10.363.4.7 operator<=> [2/3] . . . 754
10.363.4.8 operator<=> [3/3] . . . 755
10.363.5 Member Data Documentation . . . 755
10.363.5.1 m_value . . . 755
10.364 serif::mapping::PhysicalPositionFunctionCoefficient Class Reference . . . 756
10.364.1 Member Typedef Documentation . . . 757
10.364.1.1 Func . . . 757
10.364.2 Constructor & Destructor Documentation . . . 757
10.364.2.1 PhysicalPositionFunctionCoefficient() . . . 757
10.364.3 Member Function Documentation . . . 757
10.364.3.1 Eval() . . . 757
10.364.4 Member Data Documentation . . . 757
10.364.4.1 m_f . . . 757
10.364.4.2 m_mapping . . . 757
10.365 serif::dimensions::PhysicalQuantity Struct Reference . . . 758
10.365.1 Detailed Description . . . 759
10.366 serif::operators::backend::mfem::volume_keys::PhysicalValues< Contribution, OperandT > Struct Template Reference . . . 759
10.367 serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT > Struct Template Reference . . . 759
10.367.1 Detailed Description . . . 759
10.367.2 Member Typedef Documentation . . . 760
10.367.2.1 Border . . . 760
10.367.3 Member Data Documentation . . . 760
10.367.3.1 border_binding . . . 760
10.367.3.2 border_coefficient . . . 760
10.367.3.3 nodal . . . 760
10.368 serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT > Struct Template Reference . . . 760
10.368.1 Detailed Description . . . 761
10.368.2 Member Typedef Documentation . . . 761
10.368.2.1 BorderBinding . . . 761
10.368.2.2 Equation . . . 761
10.368.3 Member Data Documentation . . . 761

10.368.3.1 border_binding . . . 761
10.368.3.2 nodal . . . 761
10.369 serif::equations::PointLinearization< OperandsT, ResultT > Struct Template Reference . . . 762
10.369.1 Member Data Documentation . . . 762
10.369.1.1 field_derivatives . . . 762
10.369.1.2 position_derivative . . . 762
10.369.1.3 residual . . . 763
10.370 serif::equations::PoissonBalance Class Reference . . . 763
10.370.1 Member Typedef Documentation . . . 764
10.370.1.1 Test . . . 764
10.370.2 Constructor & Destructor Documentation . . . 764
10.370.2.1 PoissonBalance() . . . 764
10.370.3 Member Function Documentation . . . 764
10.370.3.1 contributions() . . . 764
10.370.4 Member Data Documentation . . . 764
10.370.4.1 m_four_pi_G . . . 764
10.370.4.2 name . . . 764
10.371 serif::discretization::quadrature::Policy Class Reference . . . 765
10.371.1 Constructor & Destructor Documentation . . . 765
10.371.1.1 Policy() . . . 765
10.371.2 Member Function Documentation . . . 765
10.371.2.1 compute_base_order() . . . 765
10.371.2.2 get_role_control() . . . 766
10.371.2.3 resolve() . . . 766
10.371.3 Member Data Documentation . . . 766
10.371.3.1 m_ruleSet . . . 766
10.372 serif::eos::models::Polytrope Class Reference . . . 766
10.372.1 Member Typedef Documentation . . . 767
10.372.1.1 Relations . . . 767
10.372.2 Constructor & Destructor Documentation . . . 767
10.372.2.1 Polytrope() [1/2] . . . 767
10.372.2.2 Polytrope() [2/2] . . . 767
10.372.3 Member Function Documentation . . . 768
10.372.3.1 enthalpy_scale() . . . 768
10.372.3.2 evaluate() [1/5] . . . 768
10.372.3.3 evaluate() [2/5] . . . 768
10.372.3.4 evaluate() [3/5] . . . 768
10.372.3.5 evaluate() [4/5] . . . 768
10.372.3.6 evaluate() [5/5] . . . 768
10.372.3.7 partial_derivative() [1/3] . . . 768

10.372.3.8 partial_derivative() [2/3] . . . 769
10.372.3.9 partial_derivative() [3/3] . . . 769
10.372.3.10 polytropic_constant() . . . 769
10.372.3.11 polytropic_index() . . . 769
10.372.4 Member Data Documentation . . . 769
10.372.4.1 m_enthalpy_scale . . . 769
10.372.4.2 m_polytropic_constant . . . 769
10.372.4.3 m_polytropic_index . . . 769
10.373 serif::eos::models::PolytropeParameters Struct Reference . . . 770
10.373.1 Member Data Documentation . . . 770
10.373.1.1 K . . . 770
10.373.1.2 n . . . 770
10.374 serif::discretization::blocks::gravity::potential Struct Reference . . . 770
10.374.1 Detailed Description . . . 772
10.374.2 Member Typedef Documentation . . . 772
10.374.2.1 FunctionalSpace . . . 772
10.374.2.2 Quantity . . . 773
10.374.2.3 Support . . . 773
10.374.3 Member Data Documentation . . . 773
10.374.3.1 name . . . 773
10.374.3.2 symbol . . . 773
10.375 serif::equations::gravity_contributions::PotentialValue Struct Reference . . . 774
10.375.1 Member Typedef Documentation . . . 775
10.375.1.1 Operands . . . 775
10.375.2 Member Function Documentation . . . 775
10.375.2.1 additional_order() . . . 775
10.375.2.2 evaluate() . . . 775
10.376 serif::dimensions::Power Struct Reference . . . 776
10.376.1 Detailed Description . . . 777
10.376.2 Member Data Documentation . . . 777
10.376.2.1 identifier . . . 777
10.377 serif::geometry::deformation::PowerLawRadialExtension Struct Reference . . . 777
10.377.1 Detailed Description . . . 777
10.377.2 Member Function Documentation . . . 777
10.377.2.1 operator==() . . . 777
10.377.3 Member Data Documentation . . . 778
10.377.3.1 radial_power . . . 778
10.378 serif::utils::dependencies::detail::UsedResource< GraphT >::Predicate< ResourceT > Struct Template Reference . . . 778
10.379 serif::utils::dependencies::PreparationFailure< ErrorT > Struct Template Reference . . . 779

10.379.1 Member Data Documentation . . . 779
10.379.1.1 computation_index . . . 779
10.379.1.2 error . . . 780
10.380 serif::operators::backend::mfem::PreparationRejection Struct Reference . . . 780
10.380.1 Member Data Documentation . . . 781
10.380.1.1 element . . . 781
10.380.1.2 mapping_status . . . 781
10.380.1.3 message . . . 781
10.380.1.4 quadrature_point . . . 781
10.380.1.5 reason . . . 781
10.381 serif::utils::dependencies::PreparationReport< ResultsT, SourcesT > Struct Template Reference . . . 781
10.381.1 Member Function Documentation . . . 782
10.381.1.1 computed_count() . . . 782
10.381.1.2 reused_count() . . . 782
10.381.1.3 source_changed() . . . 782
10.381.1.4 status() . . . 783
10.381.2 Member Data Documentation . . . 783
10.381.2.1 changed_sources . . . 783
10.381.2.2 statuses . . . 783
10.382 serif::utils::misc::concepts::Prepend< T, List > Struct Template Reference . . . 783
10.383 serif::utils::misc::concepts::Prepend< T, types::TypeList< Ts... > > Struct Template Reference 783
10.383.1 Member Typedef Documentation . . . 784
10.383.1.1 Type . . . 784
10.384 serif::dimensions::Pressure Struct Reference . . . 784
10.384.1 Detailed Description . . . 785
10.384.2 Member Data Documentation . . . 785
10.384.2.1 identifier . . . 785
10.385 serif::geometry::surface::keys::PressureCondition Struct Reference . . . 785
10.386 serif::geometry::surface::detail::PressureConditionTracking Struct Reference . . . 786
10.386.1 Member Function Documentation . . . 786
10.386.1.1 capture() . . . 786
10.386.1.2 identity() . . . 786
10.386.1.3 matches() . . . 786
10.387 serif::equations::PressureForce< EOST > Class Template Reference . . . 787
10.387.1 Detailed Description . . . 788
10.387.2 Member Typedef Documentation . . . 788
10.387.2.1 Enthalpy . . . 788
10.387.2.2 EnthalpyValue . . . 788
10.387.2.3 Operands . . . 788

10.387.3 Constructor & Destructor Documentation . . . 789
10.387.3.1 PressureForce() [1/2] . . . 789
10.387.3.2 PressureForce() [2/2] . . . 789
10.387.4 Member Function Documentation . . . 789
10.387.4.1 additional_order() . . . 789
10.387.4.2 evaluate() . . . 789
10.387.5 Member Data Documentation . . . 789
10.387.5.1 m_enthalpy_power . . . 789
10.387.5.2 m_eos . . . 790
10.387.5.3 name . . . 790
10.388 serif::equations::PressureSurfaceConstraint< EOST > Class Template Reference . . . 790
10.388.1 Detailed Description . . . 791
10.388.2 Member Typedef Documentation . . . 791
10.388.2.1 Enthalpy . . . 791
10.388.2.2 Operands . . . 792
10.388.3 Constructor & Destructor Documentation . . . 792
10.388.3.1 PressureSurfaceConstraint() . . . 792
10.388.4 Member Function Documentation . . . 792
10.388.4.1 evaluate() . . . 792
10.388.4.2 target_pressure() . . . 792
10.388.5 Member Data Documentation . . . 792
10.388.5.1 m_target_enthalpy . . . 792
10.388.5.2 m_target_pressure . . . 792
10.388.5.3 name . . . 793
10.388.5.4 position_independent . . . 793
10.389 serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier > Struct Template Reference . . . 793
10.389.1 Detailed Description . . . 794
10.389.2 Member Typedef Documentation . . . 794
10.389.2.1 CarrierQuantity . . . 794
10.389.2.2 CarrierTerm . . . 794
10.389.2.3 EOSStateQuantities . . . 794
10.389.2.4 EOSStateTerms . . . 794
10.389.2.5 EquationOfState . . . 794
10.389.2.6 Operands . . . 795
10.389.2.7 PhysicalCondition . . . 795
10.389.2.8 PhysicalQuantity . . . 795
10.389.2.9 Relation . . . 795
10.389.2.10 ResidualQuantity . . . 795
10.389.2.11 StateBindings . . . 795

10.389.2.12 StateQuantities . . . 795
10.389.2.13 StateTerms . . . 796
10.389.2.14 Support . . . 796
10.389.2.15 Test . . . 796
10.389.3 Member Data Documentation . . . 796
10.389.3.1 resolves_carrier . . . 796
10.390 serif::geometry::surface::PressureSurfaceDescriptor Struct Reference . . . 796
10.390.1 Member Data Documentation . . . 797
10.390.1.1 target_pressure . . . 797
10.391 serif::geometry::surface::PressureSurfaceFormulation< EOS, Bindings, Carrier > Struct Template Reference . . . 797
10.391.1 Detailed Description . . . 798
10.392 serif::discretization::backend::mfem::TensorBasis::Product Struct Reference . . . 799
10.392.1 Detailed Description . . . 799
10.392.2 Member Data Documentation . . . 799
10.392.2.1 dofs . . . 799
10.392.2.2 factors . . . 800
10.392.2.3 output . . . 800
10.393 serif::utils::dependencies::ProjectedSnapshot< FunctionT > Class Template Reference . . . 800
10.393.1 Constructor & Destructor Documentation . . . 800
10.393.1.1 ProjectedSnapshot() . . . 800
10.393.2 Member Function Documentation . . . 801
10.393.2.1 capture() . . . 801
10.393.2.2 matches() . . . 801
10.393.3 Member Data Documentation . . . 801
10.393.3.1 m_function . . . 801
10.394 serif::discretization::forms::projection< TrialT, TestT > Struct Template Reference . . . 801
10.394.1 Detailed Description . . . 803
10.394.2 Member Typedef Documentation . . . 803
10.394.2.1 operand_types . . . 803
10.394.2.2 test_type . . . 803
10.394.2.3 trial_type . . . 803
10.394.3 Member Data Documentation . . . 804
10.394.3.1 name . . . 804
10.395 serif::operators::backend::mfem::volume_keys::QuadratureLinearization< Contribution > Struct Template Reference . . . 804
10.396 serif::discretization::quadrature::QuadratureManifestOptions Struct Reference . . . 804
10.396.1 Member Data Documentation . . . 804
10.396.1.1 enabled . . . 804
10.396.1.2 include_repeated_queries . . . 805

10.396.1.3 output_file . . . 805
10.397 serif::discretization::quadrature::QuadratureOptions Struct Reference . . . 805
10.397.1 Member Data Documentation . . . 806
10.397.1.1 center_of_mass . . . 806
10.397.1.2 centrifugal . . . 806
10.397.1.3 convergence_boosts . . . 806
10.397.1.4 density_projection . . . 806
10.397.1.5 eos_closure . . . 806
10.397.1.6 error_norm . . . 807
10.397.1.7 fallback_fixed_order . . . 807
10.397.1.8 global_boost . . . 807
10.397.1.9 gravitational_energy . . . 807
10.397.1.10 gravity_boundary . . . 807
10.397.1.11 gravity_divergence . . . 807
10.397.1.12 gravity_force . . . 807
10.397.1.13 gravity_hdiv_mass . . . 807
10.397.1.14 gravity_source . . . 807
10.397.1.15 hydrostatic_equilibrium . . . 807
10.397.1.16 isobaric_surface . . . 808
10.397.1.17 manifest . . . 808
10.397.1.18 mass_conservation . . . 808
10.397.1.19 mass_normalization . . . 808
10.397.1.20 mesh_extension . . . 808
10.397.1.21 mode . . . 808
10.397.1.22 pressure_force . . . 808
10.397.1.23 pressure_integral . . . 808
10.397.1.24 quadrupole . . . 808
10.397.1.25 roles . . . 808
10.397.1.26 validation . . . 809
10.397.1.27 virial . . . 809
10.398 serif::discretization::quadrature::QuadratureRoleOptions Struct Reference . . . 809
10.398.1 Member Data Documentation . . . 810
10.398.1.1 diagnostic . . . 810
10.398.1.2 discretization . . . 810
10.398.1.3 preconditioner . . . 810
10.398.1.4 projection . . . 810
10.399 serif::discretization::quadrature::QuadratureTermOptions Struct Reference . . . 810
10.399.1 Member Data Documentation . . . 810
10.399.1.1 additional_boost . . . 810
10.399.1.2 fixed_order . . . 811

10.400 serif::discretization::quadrature::QuadratureValidationOptions Struct Reference 811
10.400.1 Member Data Documentation 811
10.400.1.1 reject_negative_boosts 811
10.400.1.2 report_unused_overrides 811
10.400.1.3 require_explicit_base_order 811
10.400.1.4 require_explicit_mfem_rule 811
10.401 serif::dimensions::QuantityAt< Index, Quantities > Struct Template Reference 812
10.402 serif::dimensions::QuantityAt< Index, QuantityList< Quantities... > > Struct Template Reference 812
10.402.1 Member Typedef Documentation 812
10.402.1.1 Type 812
10.403 serif::dimensions::QuantityList< Quantities > Struct Template Reference 812
10.404 serif::dimensions::QuantityOf< Candidate > Struct Template Reference 812
10.404.1 Detailed Description 812
10.405 serif::dimensions::QuantityOf< QuantityValue< Quantity > > Struct Template Reference . 813
10.405.1 Member Typedef Documentation 813
10.405.1.1 Type 813
10.406 serif::dimensions::QuantityValue< Quantity > Class Template Reference 813
10.406.1 Detailed Description 814
10.406.2 Constructor & Destructor Documentation 814
10.406.2.1 QuantityValue() 814
10.406.3 Member Function Documentation 815
10.406.3.1 value() 815
10.406.4 Friends And Related Symbol Documentation 815
10.406.4.1 operator* [1/2] 815
10.406.4.2 operator* [2/2] 815
10.406.4.3 operator+ 816
10.406.4.4 operator- 816
10.406.4.5 operator/ 817
10.406.4.6 operator<=> [1/3] 818
10.406.4.7 operator<=> [2/3] 818
10.406.4.8 operator<=> [3/3] 819
10.406.4.9 operator== 819
10.406.5 Member Data Documentation 820
10.406.5.1 m_value 820
10.407 serif::discretization::quadrature::Query Struct Reference 820
10.407.1 Member Data Documentation 820
10.407.1.1 base_order 820
10.407.1.2 coefficient_order 821
10.407.1.3 domain 821

10.407.1.4 form . . . 821
10.407.1.5 geometry_weight_order . . . 821
10.407.1.6 mapping . . . 821
10.407.1.7 role . . . 821
10.407.1.8 test_order . . . 821
10.407.1.9 test_order_reduction . . . 821
10.407.1.10 trial_order . . . 821
10.407.1.11 trial_order_reduction . . . 822
10.408 serif::geometry::deformation::RadialExtension Class Reference . . . 822
10.408.1 Detailed Description . . . 823
10.408.2 Member Typedef Documentation . . . 823
10.408.2.1 InputSpace . . . 823
10.408.2.2 OutputSpace . . . 823
10.408.3 Constructor & Destructor Documentation . . . 824
10.408.3.1 RadialExtension() . . . 824
10.408.4 Member Function Documentation . . . 824
10.408.4.1 apply() . . . 824
10.408.4.2 apply_transpose() . . . 824
10.408.4.3 input_size() . . . 824
10.408.4.4 interpolation() . . . 824
10.408.4.5 output_size() . . . 825
10.408.4.6 radial_weights() . . . 825
10.408.5 Friends And Related Symbol Documentation . . . 825
10.408.5.1 make_radial_extension . . . 825
10.408.6 Member Data Documentation . . . 825
10.408.6.1 is_linear . . . 825
10.408.6.2 m_interpolation . . . 825
10.408.6.3 m_radial_weights . . . 825
10.409 serif::mapping::compactification::KelvinCompactification::RadialFactors Struct Reference . . . 826
10.409.1 Member Data Documentation . . . 826
10.409.1.1 compactification_coordinate . . . 826
10.409.1.2 computational_radius . . . 826
10.409.1.3 scale . . . 826
10.409.1.4 scale_derivative . . . 826
10.410 serif::geometry::deformation::keys::RadialInterpolation Struct Reference . . . 826
10.411 serif::geometry::deformation::RadialInterpolation Struct Reference . . . 827
10.411.1 Detailed Description . . . 827
10.411.2 Member Function Documentation . . . 828
10.411.2.1 scalar_dof_count() . . . 828
10.411.2.2 surface_node_count() . . . 828

10.411.3 Member Data Documentation . . . 828
10.411.3.1 infinity_radius . . . 828
10.411.3.2 interpolation_weights . . . 828
10.411.3.3 logical_radii . . . 828
10.411.3.4 row_offsets . . . 828
10.411.3.5 stellar_radius . . . 828
10.411.3.6 stellar_support . . . 828
10.411.3.7 surface_columns . . . 829
10.411.3.8 surface_scalar_dofs . . . 829
10.411.3.9 vacuum_support . . . 829
10.412 serif::discretization::functional_space::Rank< R > Struct Template Reference . . . 829
10.412.1 Detailed Description . . . 829
10.412.2 Member Data Documentation . . . 830
10.412.2.1 value . . . 830
10.413 serif::geometry::deformation::backend::mfem::ReferenceDiscretization Struct Reference . . . 830
10.413.1 Detailed Description . . . 831
10.413.2 Member Data Documentation . . . 831
10.413.2.1 displacement_space . . . 831
10.413.2.2 infinity_attribute . . . 831
10.413.2.3 logical_mesh . . . 831
10.413.2.4 scalar_space . . . 831
10.413.2.5 stellar_attributes . . . 831
10.413.2.6 surface_attribute . . . 831
10.413.2.7 vacuum_attributes . . . 831
10.414 serif::geometry::deformation::keys::ReferenceDiscretization Struct Reference . . . 832
10.415 serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot Struct Reference . . . 832
10.415.1 Detailed Description . . . 832
10.415.2 Member Function Documentation . . . 833
10.415.2.1 operator==() . . . 833
10.415.3 Member Data Documentation . . . 833
10.415.3.1 infinity_attribute . . . 833
10.415.3.2 meshes . . . 833
10.415.3.3 spaces . . . 833
10.415.3.4 stellar_attributes . . . 833
10.415.3.5 surface_attribute . . . 833
10.415.3.6 vacuum_attributes . . . 833
10.416 serif::geometry::deformation::backend::mfem::ReferenceDiscretizationTracking Struct Reference . . . 834
10.416.1 Detailed Description . . . 834

10.416.2 Member Function Documentation . . . 834
10.416.2.1 capture() . . . 834
10.416.2.2 identity() . . . 834
10.416.2.3 matches() . . . 834
10.417 serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot Struct Reference . . . 835
10.417.1 Detailed Description . . . 835
10.417.2 Member Function Documentation . . . 836
10.417.2.1 operator==() . . . 836
10.417.3 Member Data Documentation . . . 836
10.417.3.1 boundary_attributes . . . 836
10.417.3.2 element_attributes . . . 836
10.417.3.3 identity . . . 836
10.417.3.4 nodes_sequence . . . 836
10.417.3.5 sequence . . . 836
10.417.3.6 vertices . . . 836
10.418 serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot Struct Reference . . . 837
10.418.1 Detailed Description . . . 837
10.418.2 Member Function Documentation . . . 837
10.418.2.1 operator==() . . . 837
10.418.3 Member Data Documentation . . . 837
10.418.3.1 collection . . . 837
10.418.3.2 identity . . . 837
10.418.3.3 ordering . . . 838
10.418.3.4 sequence . . . 838
10.418.3.5 size . . . 838
10.418.3.6 vector_dimension . . . 838
10.419 serif::geometry::deformation::keys::ReferenceSurface Struct Reference . . . 838
10.420 serif::geometry::deformation::ReferenceSurface Struct Reference . . . 839
10.420.1 Detailed Description . . . 839
10.420.2 Member Data Documentation . . . 839
10.420.2.1 positions . . . 839
10.420.2.2 scalar_dofs . . . 840
10.421 serif::operators::backend::mfem::volume_keys::ReferenceValues< Contribution, OperandT > Struct Template Reference . . . 840
10.422 serif::eos::relations::Relation< Output, Inputs > Struct Template Reference . . . 840
10.422.1 Member Typedef Documentation . . . 840
10.422.1.1 InputQuantities . . . 840
10.422.1.2 OutputQuantity . . . 840
10.422.2 Member Data Documentation . . . 841
10.422.2.1 inputCount . . . 841

10.423 serif::eos::relations::RelationCatalog< Relations > Struct Template Reference 841
10.423.1 Member Data Documentation 841
10.423.1.1 size 841
10.424 serif::eos::relations::RelationContainsInput< RelationT, Quantity > Struct Template Reference 842
10.425 serif::eos::relations::RelationContainsInput< Relation< Output, Inputs... >, Quantity > Struct Template Reference 843
10.426 serif::discretization::domain::relation::lists::RelationList< RelationTs > Struct Template Reference 844
10.426.1 Member Data Documentation 844
10.426.1.1 count 844
10.427 serif::discretization::domain::schema::RelationList< RelationTs > Struct Template Reference 844
10.427.1 Member Data Documentation 844
10.427.1.1 count 844
10.428 serif::geometry::surface::detail::RelationQuantities< Relation > Struct Template Reference . 845
10.429 serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > > Struct Template Reference 845
10.429.1 Member Typedef Documentation 845
10.429.1.1 AllInputs 845
10.429.1.2 NonPressureInputs 845
10.430 serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, RelationListT > Struct Template Reference 846
10.431 serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, lists::RelationList< RelationTs... > > Struct Template Reference . . . 846
10.432 serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< RelationT, DomainIDsT, BoundaryIDsT > Struct Template Reference 847
10.433 serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< DomainBoundary< BoundaryT, DomainTs... >, DomainIDsT, BoundaryIDsT > Struct Template Reference 847
10.434 serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< FullyConnected< DomainT >, DomainIDsT, BoundaryIDsT > Struct Template Reference 848
10.435 serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< Inscribed< InnerDomainT, OuterDomainT >, DomainIDsT, BoundaryIDsT > Struct Template Reference 849
10.436 serif::discretization::domain::relation::validation::RelationValidationResult Struct Reference 849
10.436.1 Member Function Documentation 850
10.436.1.1 operator bool() 850
10.436.1.2 valid() 850
10.436.2 Member Data Documentation 850
10.436.2.1 connectedDiagnostics 850
10.436.2.2 domainBoundaryDiagnostics 850
10.436.2.3 failure 851
10.436.2.4 inscribedDiagnostics 851
10.437 serif::discretization::domain::schema::validation::RelationValidationResult Struct Reference . 851
10.437.1 Member Function Documentation 851

10.437.1.1 operator bool() . . . 851
10.437.1.2 valid() . . . 852
10.437.2 Member Data Documentation . . . 852
10.437.2.1 connectedDiagnostics . . . 852
10.437.2.2 domainBoundaryDiagnostics . . . 852
10.437.2.3 failure . . . 852
10.437.2.4 inscribedDiagnostics . . . 852
10.438 serif::discretization::domain::schema::validation::RelationValidator< RelationT > Struct Template Reference . . . 852
10.439 serif::discretization::domain::schema::validation::RelationValidator< DomainBoundary< BoundaryT, DomainTs... > > Struct Template Reference . . . 852
10.439.1 Member Function Documentation . . . 853
10.439.1.1 validate() . . . 853
10.440 serif::discretization::domain::schema::validation::RelationValidator< FullyConnected< DomainT > > Struct Template Reference . . . 853
10.440.1 Member Function Documentation . . . 853
10.440.1.1 validate() . . . 853
10.441 serif::discretization::domain::schema::validation::RelationValidator< Inscribed< InnerT, OuterT > > Struct Template Reference . . . 854
10.441.1 Member Function Documentation . . . 854
10.441.1.1 validate() . . . 854
10.442 serif::discretization::blocks::density::mass::residual Struct Reference . . . 854
10.442.1 Detailed Description . . . 856
10.443 serif::discretization::blocks::displacement::vector::residual Struct Reference . . . 856
10.443.1 Detailed Description . . . 857
10.444 serif::discretization::blocks::enthalpy::specific::residual Struct Reference . . . 857
10.444.1 Detailed Description . . . 859
10.445 serif::discretization::blocks::global_scalar::residual Struct Reference . . . 859
10.445.1 Detailed Description . . . 860
10.445.2 Member Data Documentation . . . 860
10.445.2.1 static_block_size . . . 860
10.446 serif::discretization::blocks::gravity::gradient::residual Struct Reference . . . 861
10.446.1 Detailed Description . . . 862
10.447 serif::discretization::blocks::gravity::potential::residual Struct Reference . . . 862
10.447.1 Detailed Description . . . 864
10.448 serif::discretization::blocks::surface_deformation::shape_equilibrium::residual Struct Reference . . . 864
10.448.1 Detailed Description . . . 865
10.449 serif::discretization::blocks::residual_block< index_value > Struct Template Reference . . . 865
10.449.1 Detailed Description . . . 867
10.449.2 Member Function Documentation . . . 867

10.449.2.1 operator int() . . . 867
10.449.3 Member Data Documentation . . . 867
10.449.3.1 index . . . 867
10.450 serif::discretization::blocks::residual_block_base Struct Reference . . . 868
10.450.1 Detailed Description . . . 869
10.450.2 Member Data Documentation . . . 869
10.450.2.1 static_block_size . . . 869
10.451 serif::discretization::quadrature::Resolution Struct Reference . . . 869
10.451.1 Member Data Documentation . . . 870
10.451.1.1 base_order . . . 870
10.451.1.2 boost . . . 870
10.451.1.3 order . . . 870
10.451.1.4 used_fixed_order . . . 870
10.452 serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, Available > Struct Template Reference . . . 870
10.452.1 Member Data Documentation . . . 870
10.452.1.1 valid . . . 870
10.453 serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, true > Struct Template Reference . . . 871
10.453.1 Member Data Documentation . . . 871
10.453.1.1 valid . . . 871
10.454 serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, Done > Struct Template Reference . . . 872
10.455 serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, false > Struct Template Reference . . . 872
10.456 serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true > Struct Template Reference . . . 873
10.456.1 Member Typedef Documentation . . . 873
10.456.1.1 ResolvedBindings . . . 873
10.456.2 Member Data Documentation . . . 873
10.456.2.1 valid . . . 873
10.457 serif::utils::dependencies::ResolveKeys< BindingsT, KeysT > Struct Template Reference . . . 874
10.458 serif::utils::dependencies::ResolveKeys< BindingsT, KeyList< KeyTs... > > Struct Template Reference . . . 874
10.458.1 Member Typedef Documentation . . . 874
10.458.1.1 type . . . 874
10.459 serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, Valid > Struct Template Reference . . . 874
10.459.1 Member Data Documentation . . . 875
10.459.1.1 valid . . . 875
10.460 serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, true > Struct Template Reference . . . 875

10.460.1 Member Data Documentation . . . 876
10.460.1.1 valid . . . 876
10.461 serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT > Class Template Reference 876
10.461.1 Member Typedef Documentation . . . 876
10.461.1.1 Key . . . 876
10.461.1.2 Policy . . . 877
10.461.1.3 Target . . . 877
10.461.2 Constructor & Destructor Documentation . . . 877
10.461.2.1 Resource() . . . 877
10.461.3 Member Function Documentation . . . 877
10.461.3.1 get() . . . 877
10.461.3.2 policy() . . . 877
10.461.4 Member Data Documentation . . . 877
10.461.4.1 m_policy . . . 877
10.461.4.2 m_storage . . . 878
10.462 serif::utils::dependencies::detail::ResourceKeys< ResourcesT > Struct Template Reference . 878
10.463 serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > > Struct Template Reference . . . 878
10.463.1 Member Typedef Documentation . . . 878
10.463.1.1 type . . . 878
10.463.2 Member Data Documentation . . . 879
10.463.2.1 trackable . . . 879
10.464 serif::utils::dependencies::detail::Result< T > Struct Template Reference . . . 879
10.464.1 Member Typedef Documentation . . . 879
10.464.1.1 Error . . . 879
10.464.1.2 Value . . . 879
10.464.2 Member Data Documentation . . . 880
10.464.2.1 fallible . . . 880
10.464.2.2 valid . . . 880
10.465 serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > > Struct Template Reference . . . 880
10.465.1 Member Typedef Documentation . . . 880
10.465.1.1 Error [1/2] . . . 880
10.465.1.2 Error [2/2] . . . 881
10.465.1.3 Value [1/2] . . . 881
10.465.1.4 Value [2/2] . . . 881
10.465.2 Member Data Documentation . . . 881
10.465.2.1 fallible [1/2] . . . 881
10.465.2.2 fallible [2/2] . . . 881
10.465.2.3 valid [1/2] . . . 881

10.465.2.4 valid [2/2] . . . 881
10.466 serif::equations::RigidRotation Class Reference . . . 882
10.466.1 Detailed Description . . . 883
10.466.2 Member Typedef Documentation . . . 883
10.466.2.1 Matrix . . . 883
10.466.2.2 Vector . . . 883
10.466.3 Constructor & Destructor Documentation . . . 883
10.466.3.1 RigidRotation() . . . 883
10.466.4 Member Function Documentation . . . 883
10.466.4.1 angular_velocity() . . . 883
10.466.4.2 center() . . . 884
10.466.4.3 gradient() . . . 884
10.466.4.4 hessian() . . . 884
10.466.4.5 potential() . . . 884
10.466.4.6 unit_rotation() . . . 884
10.466.5 Member Data Documentation . . . 884
10.466.5.1 m_angular_velocity . . . 884
10.466.5.2 m_center . . . 884
10.466.5.3 m_hessian . . . 885
10.467 serif::discretization::quadrature::RoleControls Struct Reference . . . 885
10.467.1 Member Data Documentation . . . 885
10.467.1.1 diagnostic . . . 885
10.467.1.2 discretization . . . 886
10.467.1.3 preconditioner . . . 886
10.467.1.4 projection . . . 886
10.468 serif::discretization::blocks::rotation Struct Reference . . . 886
10.468.1 Detailed Description . . . 887
10.468.2 Member Typedef Documentation . . . 887
10.468.2.1 Terms . . . 887
10.468.3 Member Data Documentation . . . 887
10.468.3.1 angular_velocity_term . . . 887
10.468.3.2 name . . . 888
10.469 serif::equations::RotationalForce Class Reference . . . 888
10.469.1 Detailed Description . . . 889
10.469.2 Member Typedef Documentation . . . 889
10.469.2.1 AngularVelocity . . . 889
10.469.2.2 AngularVelocityValue . . . 889
10.469.2.3 Density . . . 889
10.469.2.4 DensityValue . . . 890
10.469.2.5 Operands . . . 890

10.469.2.6 Result . . . 890
10.469.3 Constructor & Destructor Documentation . . . 890
10.469.3.1 RotationalForce() . . . 890
10.469.4 Member Function Documentation . . . 890
10.469.4.1 additional_order() [1/2] . . . 890
10.469.4.2 additional_order() [2/2] . . . 890
10.469.4.3 evaluate() . . . 890
10.469.5 Member Data Documentation . . . 891
10.469.5.1 m_rotation . . . 891
10.469.5.2 name . . . 891
10.470 serif::discretization::quadrature::RuleControl Struct Reference . . . 891
10.470.1 Member Data Documentation . . . 891
10.470.1.1 boost . . . 891
10.470.1.2 fixed_order . . . 891
10.471 serif::discretization::quadrature::RuleSet Class Reference . . . 892
10.471.1 Member Function Documentation . . . 892
10.471.1.1 begin() [1/2] . . . 892
10.471.1.2 begin() [2/2] . . . 892
10.471.1.3 end() [1/2] . . . 893
10.471.1.4 end() [2/2] . . . 893
10.471.1.5 fallback_control() . . . 893
10.471.1.6 find_form_control() . . . 893
10.471.1.7 role_control() . . . 893
10.471.1.8 set_fallback_control() . . . 893
10.471.1.9 set_form_control() [1/2] . . . 893
10.471.1.10 set_form_control() [2/2] . . . 893
10.471.1.11 set_role_control() . . . 894
10.471.2 Member Data Documentation . . . 894
10.471.2.1 m_diagnosticControl . . . 894
10.471.2.2 m_discretizationControl . . . 894
10.471.2.3 m_fallbackControl . . . 894
10.471.2.4 m_formRules . . . 894
10.471.2.5 m_preconditionerControl . . . 894
10.471.2.6 m_projectionControl . . . 894
10.472 serif::eos::runtime::RuntimeCatalogDispatch< EOS, CatalogT > Struct Template Reference 894
10.473 serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > > Struct Template Reference . . . 895
10.473.1 Member Function Documentation . . . 895
10.473.1.1 evaluate() . . . 895
10.473.1.2 partial_derivative() . . . 895

10.474 serif::eos::runtime::RuntimeCatalogIsSupported< CatalogT > Struct Template Reference . 896
10.475 serif::eos::runtime::RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > > Struct Template Reference . . . 897
10.476 serif::eos::runtime::RuntimeCatalogStorage< EOS, CatalogT > Struct Template Reference . 897
10.477 serif::eos::runtime::RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > > Struct Template Reference . . . 898
10.477.1 Member Data Documentation . . . 898
10.477.1.1 descriptors . . . 898
10.478 serif::dimensions::runtime::RuntimeQuantityValue Struct Reference . . . 899
10.478.1 Detailed Description . . . 899
10.478.2 Member Data Documentation . . . 900
10.478.2.1 id . . . 900
10.478.2.2 value . . . 900
10.479 serif::dimensions::runtime::RuntimeRelationDescriptor Struct Reference . . . 900
10.479.1 Member Function Documentation . . . 900
10.479.1.1 hasPartialDerivative() . . . 900
10.479.2 Member Data Documentation . . . 901
10.479.2.1 inputQuantities . . . 901
10.479.2.2 outputQuantity . . . 901
10.479.2.3 partialDerivativeMask . . . 901
10.480 serif::eos::runtime::RuntimeRelationQuantities< RelationT > Struct Template Reference . . 901
10.481 serif::eos::runtime::RuntimeRelationQuantities< relations::Relation< OutputT, InputTs... > > Struct Template Reference . . . 901
10.481.1 Member Typedef Documentation . . . 902
10.481.1.1 Type . . . 902
10.482 serif::eos::runtime::RuntimeRelationsAreSupported< QuantitiesAreIdentified, RelationTs > Struct Template Reference . . . 902
10.483 serif::eos::runtime::RuntimeRelationsAreSupported< true, RelationTs... > Struct Template Reference . . . 903
10.484 serif::eos::runtime::RuntimeRelationStorage< EOS, RelationT > Struct Template Reference 904
10.485 serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > > Struct Template Reference . . . 904
10.485.1 Member Typedef Documentation . . . 905
10.485.1.1 RelationType . . . 905
10.485.2 Member Function Documentation . . . 905
10.485.2.1 makePartialDerivativeMask() . . . 905
10.485.3 Member Data Documentation . . . 905
10.485.3.1 descriptor . . . 905
10.485.3.2 inputQuantityIDs . . . 906
10.485.3.3 partialDerivativeMask . . . 906
10.486 serif::utils::dependencies::keys::Scale Struct Reference . . . 906

10.486.1 Detailed Description . . . 906
10.487 serif::discretization::domain::schema::validation::SchemaRelationValidationResult Struct Reference . . . 906
10.487.1 Member Function Documentation . . . 907
10.487.1.1 operator bool() . . . 907
10.487.1.2 valid() . . . 907
10.487.2 Member Data Documentation . . . 907
10.487.2.1 relationIndex . . . 907
10.487.2.2 relationName . . . 907
10.487.2.3 result . . . 907
10.488 serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, RelationListT > Struct Template Reference . . . 908
10.489 serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, relation::lists::RelationList< RelationTs... > > Struct Template Reference . . . 908
10.489.1 Member Function Documentation . . . 908
10.489.1.1 validate() . . . 908
10.490 serif::discretization::domain::schema::validation::SchemaValidationResult Struct Reference . . . 908
10.490.1 Member Function Documentation . . . 909
10.490.1.1 failed_relation_count() . . . 909
10.490.1.2 first_failed_relation_index() . . . 909
10.490.1.3 operator bool() . . . 909
10.490.1.4 passed_relation_count() . . . 909
10.490.1.5 relation_count() . . . 909
10.490.1.6 valid() . . . 909
10.490.2 Member Data Documentation . . . 910
10.490.2.1 relationResults . . . 910
10.491 serif::operators::backend::mfem::SelectedNodalValues Struct Reference . . . 910
10.491.1 Detailed Description . . . 911
10.491.2 Member Data Documentation . . . 911
10.491.2.1 selected_rows . . . 911
10.491.2.2 values . . . 911
10.492 serif::utils::errors::serif_error_printer< Message, Code > Struct Template Reference . . . 911
10.493 serif::discretization::blocks::surface_deformation::shape_equilibrium Struct Reference . . . 912
10.493.1 Detailed Description . . . 913
10.493.2 Member Data Documentation . . . 914
10.493.2.1 name . . . 914
10.494 serif::geometry::deformation::keys::ShapeCoefficients Struct Reference . . . 914
10.495 serif::geometry::deformation::spaces::ShapeParameters Struct Reference . . . 914
10.495.1 Detailed Description . . . 914
10.496 serif::discretization::forms::source< TrialT, TestT > Struct Template Reference . . . 915
10.496.1 Detailed Description . . . 916

10.496.2 Member Typedef Documentation . . . 916
10.496.2.1 operand_types . . . 916
10.496.2.2 test_type . . . 916
10.496.2.3 trial_type . . . 917
10.496.3 Member Data Documentation . . . 917
10.496.3.1 name . . . 917
10.497 serif::utils::dependencies::detail::SourceSnapshot< ResourceT > Struct Template Reference 917
10.497.1 Member Typedef Documentation . . . 918
10.497.1.1 Captured . . . 918
10.497.2 Member Data Documentation . . . 918
10.497.2.1 captured . . . 918
10.497.2.2 identity . . . 918
10.498 serif::discretization::functional_space::SpaceDescriptor Struct Reference . . . 918
10.498.1 Detailed Description . . . 919
10.498.2 Member Data Documentation . . . 919
10.498.2.1 continuity . . . 919
10.498.2.2 family . . . 919
10.498.2.3 mapping . . . 919
10.498.2.4 trace . . . 919
10.499 serif::utils::types::SparseMatrixBase< T > Struct Template Reference . . . 920
10.499.1 Member Typedef Documentation . . . 920
10.499.1.1 Type . . . 920
10.500 serif::utils::types::SparseTensorBase< T, Rank > Struct Template Reference . . . 920
10.500.1 Member Typedef Documentation . . . 920
10.500.1.1 Type . . . 920
10.500.2 Member Data Documentation . . . 921
10.500.2.1 rank . . . 921
10.501 serif::discretization::blocks::SpatialTermTraits< TermT > Struct Template Reference . . . 921
10.501.1 Detailed Description . . . 921
10.501.2 Member Typedef Documentation . . . 922
10.501.2.1 FunctionalSpace . . . 922
10.501.2.2 Support . . . 922
10.501.3 Member Data Documentation . . . 922
10.501.3.1 family . . . 922
10.501.3.2 order . . . 922
10.501.3.3 rank . . . 922
10.502 serif::discretization::blocks::enthalpy::specific Struct Reference . . . 923
10.502.1 Detailed Description . . . 925
10.502.2 Member Typedef Documentation . . . 925
10.502.2.1 FunctionalSpace . . . 925

10.502.2.2 Quantity . . . 925
10.502.2.3 Support . . . 925
10.502.3 Member Data Documentation . . . 925
10.502.3.1 name . . . 925
10.502.3.2 symbol . . . 926
10.503 serif::dimensions::SpecificEnergy Struct Reference . . . 926
10.503.1 Detailed Description . . . 927
10.503.2 Member Data Documentation . . . 927
10.503.2.1 identifier . . . 927
10.504 serif::dimensions::SpecificEnthalpy Struct Reference . . . 928
10.504.1 Detailed Description . . . 929
10.504.2 Member Data Documentation . . . 929
10.504.2.1 identifier . . . 929
10.505 serif::dimensions::SpecificEntropy Struct Reference . . . 929
10.505.1 Detailed Description . . . 930
10.505.2 Member Data Documentation . . . 930
10.505.2.1 identifier . . . 930
10.506 serif::dimensions::SpecificInternalEnergy Struct Reference . . . 931
10.506.1 Detailed Description . . . 932
10.506.2 Member Data Documentation . . . 932
10.506.2.1 identifier . . . 932
10.507 serif::geometry::deformation::SphericalHarmonicMode Struct Reference . . . 932
10.507.1 Detailed Description . . . 932
10.507.2 Member Function Documentation . . . 933
10.507.2.1 operator==() . . . 933
10.507.3 Member Data Documentation . . . 933
10.507.3.1 degree . . . 933
10.507.3.2 order . . . 933
10.508 serif::geometry::deformation::SphericalHarmonicSurface Struct Reference . . . 933
10.508.1 Detailed Description . . . 934
10.508.2 Member Function Documentation . . . 934
10.508.2.1 operator==() . . . 934
10.508.3 Member Data Documentation . . . 934
10.508.3.1 maximum_azimuthal_order . . . 934
10.508.3.2 maximum_degree . . . 934
10.508.3.3 reference_center . . . 934
10.509 serif::discretization::domain::StellarSurfaceBoundary Struct Reference . . . 935
10.509.1 Member Data Documentation . . . 936
10.509.1.1 name . . . 936
10.510 serif::discretization::blocks::surface_deformation Struct Reference . . . 936

10.510.1 Detailed Description . . . 937
10.510.2 Member Typedef Documentation . . . 937
10.510.2.1 Terms . . . 937
10.510.3 Member Data Documentation . . . 938
10.510.3.1 coefficients_term . . . 938
10.510.3.2 name . . . 938
10.510.3.3 shape_equilibrium_term . . . 938
10.511 serif::geometry::deformation::keys::SurfaceDisplacement Struct Reference . . . 938
10.512 serif::geometry::deformation::spaces::SurfaceDisplacement Struct Reference . . . 938
10.512.1 Detailed Description . . . 939
10.513 serif::geometry::deformation::SurfaceEmbedding Class Reference . . . 939
10.513.1 Detailed Description . . . 940
10.513.2 Member Typedef Documentation . . . 941
10.513.2.1 InputSpace . . . 941
10.513.2.2 OutputSpace . . . 941
10.513.3 Constructor & Destructor Documentation . . . 941
10.513.3.1 SurfaceEmbedding() . . . 941
10.513.4 Member Function Documentation . . . 941
10.513.4.1 apply() . . . 941
10.513.4.2 apply_transpose() . . . 941
10.513.4.3 has_nodal_parameters() . . . 941
10.513.4.4 input_size() . . . 942
10.513.4.5 modes() . . . 942
10.513.4.6 nodal_basis() . . . 942
10.513.4.7 output_size() . . . 942
10.513.4.8 radial_directions() . . . 942
10.513.4.9 reference_positions() . . . 942
10.513.4.10 reference_radii() . . . 942
10.513.4.11 surface_node_count() . . . 943
10.513.5 Friends And Related Symbol Documentation . . . 943
10.513.5.1 build_surface_embedding [1/2] . . . 943
10.513.5.2 build_surface_embedding [2/2] . . . 943
10.513.6 Member Data Documentation . . . 943
10.513.6.1 is_linear . . . 943
10.513.6.2 modes_ . . . 943
10.513.6.3 nodal_basis_ . . . 943
10.513.6.4 radial_directions_ . . . 943
10.513.6.5 reference_positions_ . . . 944
10.513.6.6 reference_radii_ . . . 944
10.514 serif::geometry::surface::SurfaceEOSResponse< Formulation > Struct Template Reference . . . 944

10.514.1 Detailed Description . . . 944
10.514.2 Member Data Documentation . . . 945
10.514.2.1 field_derivatives . . . 945
10.514.2.2 value . . . 945
10.515 serif::geometry::surface::SurfaceFailure Struct Reference . . . 945
10.515.1 Detailed Description . . . 946
10.515.2 Member Data Documentation . . . 946
10.515.2.1 message . . . 946
10.515.2.2 reason . . . 946
10.515.2.3 sample . . . 946
10.516 serif::geometry::deformation::keys::SurfaceMap Struct Reference . . . 946
10.517 serif::geometry::deformation::keys::SurfaceMetric Struct Reference . . . 946
10.518 serif::geometry::deformation::keys::SurfaceModel Struct Reference . . . 947
10.519 serif::geometry::surface::detail::SurfaceOperands< Terms > Struct Template Reference . . . 947
10.520 serif::geometry::surface::detail::SurfaceOperands< utils::dependencies::KeyList< Terms... > > Struct Template Reference . . . 947
10.520.1 Member Typedef Documentation . . . 947
10.520.1.1 Type . . . 947
10.521 serif::dimensions::Temperature Struct Reference . . . 948
10.521.1 Detailed Description . . . 949
10.521.2 Member Data Documentation . . . 949
10.521.2.1 identifier . . . 949
10.522 serif::discretization::backend::mfem::TensorBasis Class Reference . . . 949
10.522.1 Detailed Description . . . 951
10.522.2 Member Function Documentation . . . 951
10.522.2.1 agrees_with() . . . 951
10.522.2.2 apply() . . . 952
10.522.2.3 components() . . . 953
10.522.2.4 contract_axis() . . . 953
10.522.2.5 dofs() . . . 953
10.522.2.6 make() . . . 954
10.522.2.7 points() . . . 954
10.522.3 Member Data Documentation . . . 954
10.522.3.1 m_components . . . 954
10.522.3.2 m_dofs . . . 955
10.522.3.3 m_points . . . 955
10.522.3.4 m_products . . . 955
10.523 serif::discretization::backend::mfem::TensorBasisCache Class Reference . . . 955
10.523.1 Detailed Description . . . 956
10.523.2 Member Function Documentation . . . 956

10.523.2.1 clear() . . . 956
10.523.2.2 get() . . . 956
10.523.2.3 size() . . . 957
10.523.3 Member Data Documentation . . . 957
10.523.3.1 m_entries . . . 957
10.524 serif::operators::backend::mfem::TensorBasisCache Class Reference . . . 957
10.524.1 Detailed Description . . . 958
10.524.2 Member Function Documentation . . . 958
10.524.2.1 clear() . . . 958
10.524.2.2 get() . . . 958
10.524.2.3 size() . . . 959
10.524.3 Member Data Documentation . . . 959
10.524.3.1 m_entries . . . 959
10.525 serif::discretization::backend::mfem::TensorWorkspace Struct Reference . . . 959
10.525.1 Detailed Description . . . 960
10.525.2 Member Data Documentation . . . 960
10.525.2.1 coefficients . . . 960
10.525.2.2 first . . . 960
10.525.2.3 second . . . 960
10.526 serif::discretization::blocks::term< FieldT > Struct Template Reference . . . 961
10.526.1 Detailed Description . . . 962
10.526.2 Member Typedef Documentation . . . 962
10.526.2.1 field_type . . . 962
10.527 serif::discretization::blocks::term_base Struct Reference . . . 962
10.527.1 Detailed Description . . . 963
10.528 serif::discretization::backend::mfem::TermBinding< TermT, SchemaT > Class Template Reference . . . 964
10.528.1 Detailed Description . . . 966
10.528.2 Member Typedef Documentation . . . 966
10.528.2.1 Schema . . . 966
10.528.2.2 Term . . . 966
10.528.3 Constructor & Destructor Documentation . . . 966
10.528.3.1 TermBinding() [1/2] . . . 966
10.528.3.2 TermBinding() [2/2] . . . 967
10.528.4 Member Function Documentation . . . 967
10.528.4.1 checked_mesh() . . . 967
10.528.4.2 element_support() . . . 967
10.529 serif::discretization::blocks::TermDescriptor Struct Reference . . . 968
10.529.1 Detailed Description . . . 969
10.529.2 Friends And Related Symbol Documentation . . . 969

10.529.2.1 operator== . . . 969
10.529.3 Member Data Documentation . . . 969
10.529.3.1 field . . . 969
10.529.3.2 name . . . 970
10.530 serif::equations::TestOperandTraits< T, typename > Struct Template Reference . . . 970
10.530.1 Member Typedef Documentation . . . 970
10.530.1.1 type . . . 970
10.531 serif::equations::TestOperandTraits< T, std::void_t< typename T::TestOperand > > Struct Template Reference . . . 970
10.531.1 Member Typedef Documentation . . . 971
10.531.1.1 type [1/2] . . . 971
10.531.1.2 type [2/2] . . . 971
10.532 serif::dimensions::ThermodynamicQuantity Struct Reference . . . 971
10.532.1 Detailed Description . . . 973
10.533 serif::dimensions::runtime::ThermodynamicQuantityID Class Reference . . . 973
10.533.1 Detailed Description . . . 974
10.533.2 Constructor & Destructor Documentation . . . 974
10.533.2.1 ThermodynamicQuantityID() . . . 974
10.533.3 Member Function Documentation . . . 975
10.533.3.1 name() . . . 975
10.533.4 Friends And Related Symbol Documentation . . . 975
10.533.4.1 operator== . . . 975
10.533.5 Member Data Documentation . . . 975
10.533.5.1 m_name . . . 975
10.534 serif::dimensions::Time Struct Reference . . . 975
10.534.1 Detailed Description . . . 976
10.534.2 Member Data Documentation . . . 976
10.534.2.1 identifier . . . 976
10.535 serif::dimensions::Torque Struct Reference . . . 977
10.535.1 Detailed Description . . . 978
10.535.2 Member Data Documentation . . . 978
10.535.2.1 identifier . . . 978
10.536 serif::utils::types::TypeList< Ts > Struct Template Reference . . . 978
10.537 serif::geometry::surface::detail::UniqueTerms< Terms > Struct Template Reference . . . 978
10.538 serif::geometry::surface::detail::UniqueTerms< utils::types::TypeList< Terms... > > Struct Template Reference . . . 978
10.538.1 Member Typedef Documentation . . . 979
10.538.1.1 Type . . . 979
10.539 serif::utils::dependencies::detail::UsedResource< GraphT > Struct Template Reference . . . 979
10.540 serif::discretization::domain::VacuumDomain Struct Reference . . . 979

10.540.1 Member Data Documentation . . . 980
10.540.1.1 name . . . 980
10.541 serif::geometry::deformation::keys::VacuumModel Struct Reference . . . 980
10.542 serif::equations::ValidContributions< RowT, TupleT > Struct Template Reference . . . 981
10.543 serif::equations::ValidContributions< RowT, std::tuple< C... > > Struct Template Reference 982
10.544 serif::discretization::forms::Operation::Value Struct Reference . . . 982
10.544.1 Detailed Description . . . 983
10.545 serif::discretization::blocks::density::mass::value Struct Reference . . . 983
10.545.1 Detailed Description . . . 984
10.546 serif::discretization::blocks::displacement::vector::value Struct Reference . . . 984
10.546.1 Detailed Description . . . 986
10.547 serif::discretization::blocks::enthalpy::specific::value Struct Reference . . . 986
10.547.1 Detailed Description . . . 987
10.548 serif::discretization::blocks::global_scalar::value Struct Reference . . . 987
10.548.1 Detailed Description . . . 989
10.548.2 Member Data Documentation . . . 989
10.548.2.1 static_block_size . . . 989
10.549 serif::discretization::blocks::gravity::gradient::value Struct Reference . . . 989
10.549.1 Detailed Description . . . 991
10.550 serif::discretization::blocks::gravity::potential::value Struct Reference . . . 991
10.550.1 Detailed Description . . . 992
10.551 serif::discretization::blocks::surface_deformation::coefficients::value Struct Reference . . . 992
10.551.1 Detailed Description . . . 994
10.552 serif::discretization::blocks::value_block< index_value > Struct Template Reference . . . 994
10.552.1 Detailed Description . . . 995
10.552.2 Member Function Documentation . . . 996
10.552.2.1 operator int() . . . 996
10.552.3 Member Data Documentation . . . 996
10.552.3.1 index . . . 996
10.553 serif::discretization::blocks::value_block_base Struct Reference . . . 996
10.553.1 Detailed Description . . . 998
10.553.2 Member Data Documentation . . . 998
10.553.2.1 static_block_size . . . 998
10.554 serif::utils::dependencies::ValueSnapshot Struct Reference . . . 998
10.554.1 Detailed Description . . . 999
10.554.2 Member Function Documentation . . . 999
10.554.2.1 capture() . . . 999
10.554.2.2 matches() . . . 1000
10.555 serif::discretization::blocks::displacement::vector Struct Reference . . . 1000
10.555.1 Detailed Description . . . 1002

10.555.2 Member Typedef Documentation . . . 1002
10.555.2.1 FunctionalSpace . . . 1002
10.555.2.2 Quantity . . . 1002
10.555.2.3 Support . . . 1002
10.555.3 Member Data Documentation . . . 1003
10.555.3.1 name . . . 1003
10.555.3.2 symbol . . . 1003
10.556 serif::utils::types::VectorBase< T > Struct Template Reference . . . 1003
10.556.1 Member Typedef Documentation . . . 1003
10.556.1.1 Type . . . 1003
10.557 serif::dimensions::Velocity Struct Reference . . . 1004
10.557.1 Detailed Description . . . 1005
10.557.2 Member Data Documentation . . . 1005
10.557.2.1 identifier . . . 1005
10.558 serif::dimensions::Volume Struct Reference . . . 1005
10.558.1 Detailed Description . . . 1006
10.558.2 Member Data Documentation . . . 1006
10.558.2.1 identifier . . . 1006
10.559 serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT > Struct Template Reference . . . 1007
10.559.1 Detailed Description . . . 1007
10.559.2 Member Typedef Documentation . . . 1007
10.559.2.1 DisplacementBinding . . . 1007
10.559.3 Member Data Documentation . . . 1008
10.559.3.1 discretization . . . 1008
10.559.3.2 equation . . . 1008
10.559.3.3 options . . . 1008
10.560 serif::operators::VolumeContributions< TupleT > Struct Template Reference . . . 1008
10.561 serif::operators::VolumeContributions< std::tuple< C... > > Struct Template Reference . . . 1009
10.562 serif::operators::backend::mfem::VolumeData< EquationT, SchemaT > Struct Template Reference . . . 1010
10.562.1 Detailed Description . . . 1011
10.562.2 Member Typedef Documentation . . . 1011
10.562.2.1 Equation . . . 1011
10.562.2.2 Plan . . . 1011
10.562.2.3 Schema . . . 1011
10.562.3 Member Data Documentation . . . 1011
10.562.3.1 bindings . . . 1011
10.562.3.2 configuration . . . 1011
10.562.3.3 contributions . . . 1011

10.562.3.4 residual . . . 1012
10.563 serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT > Struct Template Reference . . . 1012
10.563.1 Detailed Description . . . 1012
10.563.2 Member Typedef Documentation . . . 1013
10.563.2.1 DisplacementBinding . . . 1013
10.563.2.2 RowBinding . . . 1013
10.563.3 Member Data Documentation . . . 1013
10.563.3.1 compactification . . . 1013
10.563.3.2 compactification_sequence . . . 1013
10.563.3.3 compactification_space . . . 1013
10.563.3.4 displacement_binding . . . 1013
10.563.3.5 mapper . . . 1013
10.563.3.6 residual_binding . . . 1014
10.564 serif::geometry::deformation::keys::VolumeDisplacement Struct Reference . . . 1014
10.565 serif::geometry::deformation::spaces::VolumeDisplacement Struct Reference . . . 1014
10.565.1 Detailed Description . . . 1014
10.566 serif::geometry::deformation::keys::VolumeExtension Struct Reference . . . 1014
10.567 serif::mapping::VolumeMappingCache Class Reference . . . 1015
10.567.1 Member Function Documentation . . . 1015
10.567.1.1 get_dimension() . . . 1015
10.567.1.2 get_point_count() . . . 1016
10.567.1.3 get_point_data() . . . 1016
10.567.1.4 load() . . . 1016
10.567.1.5 load_inverse_jacobian() . . . 1016
10.567.1.6 set_size() . . . 1016
10.567.1.7 store() . . . 1016
10.567.1.8 validate_point() . . . 1016
10.567.2 Member Data Documentation . . . 1016
10.567.2.1 m_data . . . 1016
10.567.2.2 m_dimension . . . 1017
10.567.2.3 m_point_count . . . 1017
10.567.2.4 m_point_stride . . . 1017
10.568 serif::mapping::VolumeMappingContext Struct Reference . . . 1017
10.568.1 Member Data Documentation . . . 1018
10.568.1.1 mapping . . . 1018
10.568.1.2 quadrature . . . 1018
10.569 serif::mapping::VolumeMappingVariation Struct Reference . . . 1018
10.569.1 Member Data Documentation . . . 1019
10.569.1.1 inverse_element_jacobian_variation . . . 1019

10.569.1.2 mapping . . . 1019
10.569.1.3 weight_variation . . . 1019
10.570 serif::operators::VolumeOperands< C, ListT > Struct Template Reference . . . 1019
10.571 serif::operators::VolumeOperands< C, utils::types::TypeList< O... > > Struct Template Reference . . . 1020
10.572 serif::operators::backend::mfem::VolumeOptions Struct Reference . . . 1021
10.572.1 Detailed Description . . . 1021
10.572.2 Member Data Documentation . . . 1022
10.572.2.1 contraction_mode . . . 1022
10.572.2.2 quadrature_policy . . . 1022
10.573 serif::mapping::VolumeQuadratureContext Struct Reference . . . 1022
10.573.1 Member Data Documentation . . . 1022
10.573.1.1 J_element_to_physical_inverse . . . 1022
10.573.1.2 mapping_determinate . . . 1022
10.573.1.3 physical_integration_weight . . . 1022
10.574 serif::operators::backend::mfem::VolumeScratch Struct Reference . . . 1023
10.574.1 Detailed Description . . . 1023
10.574.2 Member Data Documentation . . . 1023
10.574.2.1 basis . . . 1023
10.574.2.2 workspace . . . 1023
10.575 serif::dimensions::WithRespectTo< Quantity > Struct Template Reference . . . 1024
10.575.1 Detailed Description . . . 1024
10.576 serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace Struct Reference . . . 1024
10.576.1 Member Data Documentation . . . 1025
10.576.1.1 extension_workspace . . . 1025
10.576.1.2 parameter_pullback . . . 1025
10.576.1.3 surface_direction . . . 1025
10.576.1.4 surface_displacement . . . 1025
10.576.1.5 surface_dual . . . 1025
10.576.1.6 surface_pullback . . . 1026
10.576.1.7 surface_workspace . . . 1026
10.577 serif::operators::backend::mfem::Workspace Struct Reference . . . 1026
10.577.1 Detailed Description . . . 1027
10.577.2 Member Data Documentation . . . 1027
10.577.2.1 basis . . . 1027
10.577.2.2 displacement_direction . . . 1027
10.577.2.3 displacement_dual . . . 1027
10.577.2.4 displacement_gradient . . . 1027
10.577.2.5 displacement_samples . . . 1027

10.577.2.6 displacement_value . . . 1027
10.577.2.7 element_result . . . 1028
10.577.2.8 element_values . . . 1028
10.577.2.9 full_result . . . 1028
10.577.2.10 mapping_context . . . 1028
10.577.2.11 mapping_variation . . . 1028
10.577.2.12 mapping_workspace . . . 1028
10.577.2.13 operand_dual . . . 1028
10.577.2.14 operand_values . . . 1028
10.577.2.15 quadrature_values . . . 1028
10.577.2.16 reference_test_values . . . 1028
10.577.2.17 test_basis_variation . . . 1029
10.577.2.18 test_values . . . 1029
10.577.2.19 test_variation_result . . . 1029

11 File Documentation 1031
11.1 docs/static/mainpage.md File Reference . . . 1031
11.2 src/compile_time_checks/serif/dimensions/quantities.cpp File Reference . . . 1031
11.3 src/compile_time_checks/serif/discretization/blocks/fields.cpp File Reference . . . 1032
11.4 src/compile_time_checks/serif/discretization/forms/forms.cpp File Reference . . . 1032
11.5 src/compile_time_checks/serif/discretization/forms/operands.cpp File Reference . . . 1033
11.6 src/compile_time_checks/serif/discretization/quadrature/backend/mfem/mfem_resolver.cpp File Reference . . . 1033
11.7 src/compile_time_checks/serif/discretization/quadrature/backend/mfem/mfem_static_order.cpp File Reference . . . 1034
11.8 src/compile_time_checks/serif/eos/evaluation.cpp File Reference . . . 1034
11.9 src/compile_time_checks/serif/eos/models/traits.cpp File Reference . . . 1035
11.10 src/compile_time_checks/serif/eos/runtime/details.cpp File Reference . . . 1036
11.11 src/compile_time_checks/serif/eos/runtime/views.cpp File Reference . . . 1037
11.11.1 Enumeration Type Documentation . . . 1038
11.11.1.1 SERiFErrorCode . . . 1038
11.12 src/compile_time_checks/serif/geometry/deformation.cpp File Reference . . . 1038
11.13 src/compile_time_checks/serif/geometry/surface.cpp File Reference . . . 1039
11.14 src/compile_time_checks/serif/operators/computation_owners.cpp File Reference . . . 1040
11.15 src/compile_time_checks/serif/operators/plan.cpp File Reference . . . 1041
11.16 src/compile_time_checks/serif/operators/serial_backend.cpp File Reference . . . 1042
11.17 src/compile_time_checks/serif/utils/dependencies/binding.cpp File Reference . . . 1043
11.18 src/compile_time_checks/serif/utils/dependencies/computation_graph.cpp File Reference . . . 1044
11.19 src/compile_time_checks/serif/utils/dependencies/computation_store.cpp File Reference . . . 1044
11.20 src/compile_time_checks/serif/utils/dependencies/dependency_context.cpp File Reference . . . 1045
11.21 src/compile_time_checks/serif/utils/dependencies/key_list.cpp File Reference . . . 1046

11.22 src/compile_time_checks/serif/utils/dependencies/mixed_errors.cpp File Reference 1047
11.23 src/include/serif/dimensions/dimensions.hpp File Reference 1048
11.23.1 Detailed Description 1051
11.24 src/include/serif/dimensions/partial.hpp File Reference 1051
11.24.1 Detailed Description 1052
11.25 src/include/serif/dimensions/quantities.hpp File Reference 1053
11.25.1 Detailed Description 1054
11.26 src/include/serif/dimensions/type_alias.hpp File Reference 1054
11.26.1 Detailed Description 1057
11.27 src/include/serif/discretization/backend/mfem/basis_evaluation.hpp File Reference 1057
11.27.1 Detailed Description 1059
11.28 src/include/serif/discretization/backend/mfem/coefficient_tracking.hpp File Reference . . . 1059
11.28.1 Detailed Description 1060
11.29 src/include/serif/discretization/backend/mfem/dof_map.hpp File Reference 1060
11.29.1 Detailed Description 1061
11.30 src/include/serif/discretization/backend/mfem/dof_selection.hpp File Reference 1062
11.30.1 Detailed Description 1063
11.31 src/include/serif/discretization/backend/mfem/mfem_backend.hpp File Reference 1063
11.31.1 Detailed Description 1064
11.32 src/include/serif/discretization/backend/mfem/point_dofs.hpp File Reference 1064
11.32.1 Detailed Description 1065
11.33 src/include/serif/discretization/backend/mfem/tensor_basis.hpp File Reference 1066
11.33.1 Detailed Description 1067
11.34 src/include/serif/discretization/backend/mfem/term_binding.hpp File Reference 1068
11.34.1 Detailed Description 1069
11.35 src/include/serif/discretization/blocks/base.hpp File Reference 1069
11.35.1 Detailed Description 1070
11.36 src/include/serif/discretization/blocks/blocks.hpp File Reference 1070
11.37 src/include/serif/discretization/blocks/fields.hpp File Reference 1071
11.37.1 Detailed Description 1074
11.38 src/include/serif/dimensions/runtime/runtime.hpp File Reference 1075
11.38.1 Detailed Description 1077
11.39 src/include/serif/discretization/blocks/runtime.hpp File Reference 1077
11.39.1 Detailed Description 1079
11.40 src/include/serif/discretization/domain/relation/validation/runtime.hpp File Reference . . . 1079
11.41 src/include/serif/discretization/forms/runtime.hpp File Reference 1081
11.41.1 Detailed Description 1083
11.42 src/include/serif/discretization/blocks/traits.hpp File Reference 1083
11.42.1 Detailed Description 1084
11.43 src/include/serif/discretization/domain/ids/lists/traits.hpp File Reference 1085

11.43.1 Detailed Description . . . 1086
11.44 src/include/serif/discretization/domain/relation/lists/traits.hpp File Reference . . . 1086
11.45 src/include/serif/discretization/domain/schema/traits.hpp File Reference . . . 1088
11.46 src/include/serif/eos/models/traits.hpp File Reference . . . 1089
11.47 src/include/serif/eos/relations/traits.hpp File Reference . . . 1091
11.48 src/include/serif/discretization/domain/ids/boundary.hpp File Reference . . . 1092
11.48.1 Detailed Description . . . 1093
11.49 src/include/serif/discretization/domain/schema/validation/boundary.hpp File Reference . . . 1094
11.50 src/include/serif/discretization/domain/ids/domain.hpp File Reference . . . 1095
11.50.1 Detailed Description . . . 1097
11.51 src/include/serif/discretization/domain/ids/lists/lists.hpp File Reference . . . 1097
11.51.1 Detailed Description . . . 1098
11.52 src/include/serif/discretization/domain/ids/validation/unique_ids.hpp File Reference . . . 1099
11.52.1 Detailed Description . . . 1100
11.53 src/include/serif/discretization/domain/mesh/topology.hpp File Reference . . . 1100
11.54 src/include/serif/discretization/domain/physical_domains.hpp File Reference . . . 1101
11.55 src/include/serif/discretization/domain/relation/lists/relation_list.hpp File Reference . . . 1102
11.56 src/include/serif/discretization/domain/relation/relations.hpp File Reference . . . 1103
11.57 src/include/serif/eos/relations/relations.hpp File Reference . . . 1105
11.58 src/include/serif/geometry/surface/detail/relations.hpp File Reference . . . 1106
11.59 src/include/serif/discretization/domain/relation/validation/validation.hpp File Reference . . . 1108
11.60 src/include/serif/discretization/domain/resolver.hpp File Reference . . . 1110
11.61 src/include/serif/discretization/domain/schema/domain_schema.hpp File Reference . . . 1111
11.62 src/include/serif/discretization/domain/schema/schemas.hpp File Reference . . . 1113
11.63 src/include/serif/discretization/domain/schema/utils.hpp File Reference . . . 1115
11.64 src/include/serif/discretization/domain/schema/validation/all.hpp File Reference . . . 1116
11.65 src/include/serif/discretization/domain/schema/validation/connected.hpp File Reference . . . 1117
11.66 src/include/serif/discretization/domain/schema/validation/inscribed.hpp File Reference . . . 1118
11.67 src/include/serif/discretization/domain/schema/validation/results.hpp File Reference . . . 1120
11.68 src/include/serif/discretization/domain/schema/validation/validator.hpp File Reference . . . 1121
11.69 src/include/serif/discretization/domain/types.hpp File Reference . . . 1123
11.70 src/include/serif/discretization/forms/concept.hpp File Reference . . . 1124
11.70.1 Detailed Description . . . 1126
11.71 src/include/serif/discretization/forms/form_traits.hpp File Reference . . . 1126
11.71.1 Detailed Description . . . 1128
11.72 src/include/serif/discretization/forms/forms.hpp File Reference . . . 1128
11.72.1 Detailed Description . . . 1130
11.73 src/include/serif/discretization/forms/operands.hpp File Reference . . . 1131
11.73.1 Detailed Description . . . 1132
11.74 src/include/serif/discretization/forms/value.hpp File Reference . . . 1132

11.74.1 Detailed Description 1134
11.75 src/include/serif/discretization/functional_space/space.hpp File Reference 1135
11.75.1 Detailed Description 1138
11.76 src/include/serif/discretization/quadrature/backend/mfem/configure.hpp File Reference . . . 1138
11.77 src/include/serif/discretization/quadrature/backend/mfem/mfem_resolver.hpp File Reference 1139
11.78 src/include/serif/discretization/quadrature/backend/mfem/mfem_static_order.hpp File Reference 1141
11.79 src/include/serif/discretization/quadrature/backend/order.hpp File Reference 1143
11.80 src/include/serif/discretization/quadrature/policy.hpp File Reference 1145
11.81 src/include/serif/discretization/quadrature/query.hpp File Reference 1146
11.82 src/include/serif/discretization/quadrature/rules.hpp File Reference 1148
11.83 src/include/serif/eos/models/polytropic.hpp File Reference 1150
11.84 src/include/serif/eos/relations/catalog.hpp File Reference 1151
11.85 src/include/serif/eos/runtime/views.hpp File Reference 1152
11.86 src/include/serif/equations/angular_momentum.hpp File Reference 1154
11.87 src/include/serif/equations/arithmetic.hpp File Reference 1155
11.88 src/include/serif/equations/barotropic_closure.hpp File Reference 1156
11.89 src/include/serif/equations/centering.hpp File Reference 1157
11.90 src/include/serif/equations/central_density.hpp File Reference 1158
11.91 src/include/serif/equations/displacement_force.hpp File Reference 1159
11.92 src/include/serif/equations/gravity_force.hpp File Reference 1160
11.93 src/include/serif/equations/hydrostatic_equilibrium.hpp File Reference 1161
11.94 src/include/serif/equations/mass_normalization.hpp File Reference 1162
11.95 src/include/serif/equations/mixed_gravity.hpp File Reference 1163
11.96 src/include/serif/equations/point.hpp File Reference 1164
11.97 src/include/serif/equations/pressure_force.hpp File Reference 1165
11.98 src/include/serif/equations/pressure_surface.hpp File Reference 1166
11.99 src/include/serif/equations/registry.hpp File Reference 1167
11.100 src/include/serif/equations/rigid_rotation.hpp File Reference 1169
11.101 src/include/serif/equations/rotational_force.hpp File Reference 1170
11.102 src/include/serif/equations/sum.hpp File Reference 1171
11.103 src/include/serif/geometry/deformation/actions.hpp File Reference 1171
11.104 src/include/serif/geometry/deformation/backend/mfem/computations.hpp File Reference . 1174
11.105 src/include/serif/geometry/deformation/computations.hpp File Reference 1175
11.106 src/include/serif/geometry/surface/computations.hpp File Reference 1176
11.107 src/include/serif/geometry/deformation/backend/mfem/detail/radial_locator.hpp File Reference 1178
11.108 src/include/serif/geometry/deformation/backend/mfem/geometry_step.hpp File Reference 1179
11.109 src/include/serif/geometry/deformation/geometry_step.hpp File Reference 1181
11.110 src/include/serif/geometry/deformation/backend/mfem/inputs.hpp File Reference 1182

11.111 src/include/serif/geometry/deformation/backend/mfem/radial_interpolation.hpp File Reference . . . 1183
11.112 src/include/serif/geometry/deformation/backend/mfem/reference_geometry.hpp File Reference . . . 1185
11.113 src/include/serif/geometry/deformation/backend/mfem/surface_metric.hpp File Reference 1186
11.114 src/include/serif/geometry/deformation/backend/mfem/volume_displacement.hpp File Reference . . . 1187
11.115 src/include/serif/geometry/deformation/composition.hpp File Reference . . . 1188
11.116 src/include/serif/geometry/deformation/deformation.hpp File Reference . . . 1190
11.117 src/include/serif/geometry/deformation/detail/action_storage.hpp File Reference . . . 1190
11.118 src/include/serif/geometry/deformation/detail/determinant_polynomial.hpp File Reference 1192
11.119 src/include/serif/geometry/deformation/detail/harmonics.hpp File Reference . . . 1193
11.120 src/include/serif/geometry/deformation/failure.hpp File Reference . . . 1194
11.121 src/include/serif/geometry/deformation/linearization.hpp File Reference . . . 1196
11.122 src/include/serif/geometry/surface/linearization.hpp File Reference . . . 1197
11.123 src/include/serif/geometry/deformation/models.hpp File Reference . . . 1199
11.124 src/include/serif/geometry/deformation/radial_extension.hpp File Reference . . . 1200
11.125 src/include/serif/geometry/deformation/reference_surface.hpp File Reference . . . 1201
11.126 src/include/serif/geometry/deformation/spaces.hpp File Reference . . . 1203
11.127 src/include/serif/geometry/deformation/surface_embedding.hpp File Reference . . . 1203
11.128 src/include/serif/geometry/surface/compilation.hpp File Reference . . . 1205
11.129 src/include/serif/geometry/surface/detail/batches.hpp File Reference . . . 1207
11.130 src/include/serif/eos/runtime/details.hpp File Reference . . . 1209
11.131 src/include/serif/geometry/surface/details.hpp File Reference . . . 1212
11.132 src/include/serif/eos/evaluation.hpp File Reference . . . 1212
11.133 src/include/serif/geometry/surface/detail/evaluation.hpp File Reference . . . 1213
11.134 src/include/serif/geometry/surface/evaluation.hpp File Reference . . . 1215
11.135 src/include/serif/geometry/surface/isobaric.hpp File Reference . . . 1216
11.136 src/include/serif/geometry/surface/samples.hpp File Reference . . . 1219
11.137 src/include/serif/geometry/surface/surface.hpp File Reference . . . 1221
11.137.1 Detailed Description . . . 1222
11.138 src/include/serif/mapping/coefficients.hpp File Reference . . . 1222
11.139 src/include/serif/mapping/compactification/compactification.hpp File Reference . . . 1223
11.140 src/include/serif/mapping/compactification/input_result.hpp File Reference . . . 1224
11.141 src/include/serif/mapping/compactification/kelvin.hpp File Reference . . . 1226
11.142 src/include/serif/utils/user_options/mapping/compactification/kelvin.hpp File Reference . 1227
11.143 src/include/serif/mapping/compactification/variation.hpp File Reference . . . 1228
11.144 src/include/serif/mapping/variation.hpp File Reference . . . 1229
11.145 src/include/serif/mapping/context.hpp File Reference . . . 1230
11.146 src/include/serif/mapping/data.hpp File Reference . . . 1231
11.147 src/include/serif/mapping/domain_mapper.hpp File Reference . . . 1232

11.148 src/include/serif/utils/user_options/mapping/domain_mapper.hpp File Reference 1233
11.149 src/include/serif/mapping/evaluator.hpp File Reference 1234
11.150 src/include/serif/mapping/status.hpp File Reference 1235
11.151 src/include/serif/mapping/transformations.hpp File Reference 1237
11.152 src/include/serif/mapping/volume_cache.hpp File Reference 1239
11.153 src/include/serif/meta/enums.hpp File Reference 1240
11.154 src/include/serif/meta/names.hpp File Reference 1242
11.155 src/include/serif/operators/backend/mfem/element_data.hpp File Reference 1243
11.156 src/include/serif/operators/backend/mfem/element_evaluation.hpp File Reference 1244
11.157 src/include/serif/operators/backend/mfem/element_preparation.hpp File Reference 1246
11.158 src/include/serif/operators/backend/mfem/element_transpose.hpp File Reference 1248
11.159 src/include/serif/operators/backend/mfem/jacobian_operator.hpp File Reference 1250
11.160 src/include/serif/operators/backend/mfem/nodal_computations.hpp File Reference 1251
11.161 src/include/serif/operators/backend/mfem/nodal_data.hpp File Reference 1252
11.162 src/include/serif/operators/backend/mfem/nodal_evaluation.hpp File Reference 1254
11.163 src/include/serif/operators/backend/mfem/nodal_inputs.hpp File Reference 1255
11.164 src/include/serif/operators/backend/mfem/nodal_rows.hpp File Reference 1257
11.165 src/include/serif/operators/backend/mfem/operand_evaluation.hpp File Reference 1259
11.166 src/include/serif/operators/backend/mfem/point_border.hpp File Reference 1261
11.167 src/include/serif/operators/backend/mfem/preparation.hpp File Reference 1262
11.168 src/include/serif/utils/dependencies/preparation.hpp File Reference 1264
11.169 src/include/serif/operators/backend/mfem/quadrature.hpp File Reference 1265
11.170 src/include/serif/operators/backend/mfem/rejection.hpp File Reference 1266
11.171 src/include/serif/operators/backend/mfem/term_coefficients.hpp File Reference 1268
11.172 src/include/serif/operators/backend/mfem/volume_application.hpp File Reference 1269
11.173 src/include/serif/operators/backend/mfem/volume_computations.hpp File Reference . . . 1271
11.174 src/include/serif/operators/backend/mfem/volume_data.hpp File Reference 1273
11.175 src/include/serif/operators/backend/mfem/volume_inputs.hpp File Reference 1276
11.176 src/include/serif/operators/backend/mfem/volume_stages.hpp File Reference 1277
11.177 src/include/serif/operators/backend/mfem/workspace.hpp File Reference 1279
11.178 src/include/serif/dimensions/runtime/concepts.hpp File Reference 1281
11.178.1 Detailed Description 1283
11.179 src/include/serif/discretization/blocks/concepts.hpp File Reference 1283
11.179.1 Detailed Description 1284
11.180 src/include/serif/discretization/domain/concepts.hpp File Reference 1285
11.181 src/include/serif/discretization/domain/ids/concepts.hpp File Reference 1286
11.181.1 Detailed Description 1288
11.182 src/include/serif/discretization/domain/ids/lists/concepts.hpp File Reference 1288
11.182.1 Detailed Description 1289
11.183 src/include/serif/discretization/domain/ids/validation/concepts.hpp File Reference 1290

11.183.1 Detailed Description 1291
11.184 src/include/serif/discretization/domain/relation/concepts.hpp File Reference 1292
11.185 src/include/serif/discretization/domain/relation/lists/concepts.hpp File Reference 1293
11.186 src/include/serif/discretization/domain/relation/validation/concepts.hpp File Reference . . 1294
11.187 src/include/serif/discretization/domain/schema/concepts.hpp File Reference 1295
11.188 src/include/serif/discretization/quadrature/backend/mfem/concepts.hpp File Reference . . 1297
11.189 src/include/serif/eos/models/concepts.hpp File Reference 1299
11.190 src/include/serif/eos/relations/concepts.hpp File Reference 1300
11.191 src/include/serif/eos/runtime/concepts.hpp File Reference 1301
11.192 src/include/serif/equations/concepts.hpp File Reference 1303
11.193 src/include/serif/operators/concepts.hpp File Reference 1304
11.194 src/include/serif/equations/contributions.hpp File Reference 1306
11.195 src/include/serif/operators/contributions.hpp File Reference 1307
11.196 src/include/serif/operators/plan.hpp File Reference 1309
11.197 src/include/serif/utils/dependencies/binding.hpp File Reference 1310
11.198 src/include/serif/utils/dependencies/computation.hpp File Reference 1312
11.199 src/include/serif/utils/dependencies/computation_cache.hpp File Reference 1314
11.200 src/include/serif/utils/dependencies/computation_graph.hpp File Reference 1316
11.201 src/include/serif/utils/dependencies/computation_store.hpp File Reference 1318
11.202 src/include/serif/utils/dependencies/dependencies.hpp File Reference 1320
11.202.1 Detailed Description 1321
11.203 src/include/serif/utils/dependencies/dependency_context.hpp File Reference 1324
11.204 src/include/serif/utils/dependencies/detail/cache_storage.hpp File Reference 1325
11.205 src/include/serif/utils/dependencies/detail/cache_traits.hpp File Reference 1327
11.206 src/include/serif/utils/dependencies/detail/graph.hpp File Reference 1329
11.207 src/include/serif/utils/dependencies/detail/result.hpp File Reference 1330
11.208 src/include/serif/utils/dependencies/detail/schema.hpp File Reference 1332
11.209 src/include/serif/utils/dependencies/detail/type_lists.hpp File Reference 1334
11.210 src/include/serif/utils/dependencies/incremental_computation.hpp File Reference 1336
11.211 src/include/serif/utils/dependencies/key_list.hpp File Reference 1338
11.212 src/include/serif/utils/dependencies/key_list_operations.hpp File Reference 1340
11.213 src/include/serif/utils/dependencies/keys.hpp File Reference 1341
11.214 src/include/serif/utils/dependencies/resource.hpp File Reference 1342
11.215 src/include/serif/utils/dependencies/snapshot.hpp File Reference 1344
11.216 src/include/serif/utils/error/codes.hpp File Reference 1345
11.217 src/include/serif/utils/error/errors.hpp File Reference 1347
11.218 src/include/serif/utils/misc/compensated_sum.hpp File Reference 1348
11.219 src/include/serif/utils/misc/concepts/concat.hpp File Reference 1349
11.220 src/include/serif/utils/misc/concepts/enumeration.hpp File Reference 1350
11.221 src/include/serif/eos/exceptions.hpp File Reference 1351

11.222 src/include/serif/utils/misc/concepts/exceptions.hpp File Reference . . . 1352
11.223 src/include/serif/utils/misc/concepts/filter.hpp File Reference . . . 1353
11.224 src/include/serif/utils/misc/concepts/numeric.hpp File Reference . . . 1354
11.225 src/include/serif/utils/checks/finite.hpp File Reference . . . 1355
11.226 src/include/serif/utils/misc/finite.hpp File Reference . . . 1356
11.227 src/include/serif/utils/misc/std_helper/cleaning.hpp File Reference . . . 1357
11.228 src/include/serif/utils/misc/std_helper/variadic_cleaning.hpp File Reference . . . 1358
11.229 src/include/serif/utils/misc/terminal/colors.hpp File Reference . . . 1359
11.230 src/include/serif/utils/misc/concepts/type_uniqueness.hpp File Reference . . . 1361
11.231 src/include/serif/utils/misc/traits/type_uniqueness.hpp File Reference . . . 1362
11.232 src/include/serif/utils/format/strings.hpp File Reference . . . 1363
11.233 src/include/serif/utils/types/strings.hpp File Reference . . . 1364
11.234 src/include/serif/utils/types/tensor/backend/mfem/tensor.hpp File Reference . . . 1365
11.235 src/include/serif/utils/types/tensor/tensor.hpp File Reference . . . 1366
11.236 src/include/serif/utils/types/type_list.hpp File Reference . . . 1367
11.237 src/include/serif/utils/warnings/warning.hpp File Reference . . . 1368
11.237.1 Macro Definition Documentation . . . 1369
11.237.1.1 DISABLE_WARNING . . . 1369
11.238 src/lib/serif/discretization/backend/mfem/basis_evaluation.cpp File Reference . . . 1370
11.239 src/lib/serif/discretization/domain/mesh/topology.cpp File Reference . . . 1370
11.240 src/lib/serif/discretization/domain/schema/validation/results.cpp File Reference . . . 1371
11.241 src/lib/serif/discretization/quadrature/policy.cpp File Reference . . . 1372
11.242 src/lib/serif/eos/models/polytropic.cpp File Reference . . . 1373
11.243 src/lib/serif/geometry/deformation/backend/mfem/geometry_step.cpp File Reference . . . 1374
11.244 src/lib/serif/geometry/deformation/backend/mfem/inputs.cpp File Reference . . . 1375
11.245 src/lib/serif/geometry/deformation/backend/mfem/radial_interpolation.cpp File Reference 1375
11.246 src/lib/serif/geometry/deformation/backend/mfem/radial_locator.cpp File Reference . . . 1376
11.247 src/lib/serif/geometry/deformation/backend/mfem/reference_geometry.cpp File Reference 1377
11.248 src/lib/serif/geometry/deformation/backend/mfem/surface_metric.cpp File Reference . . . 1378
11.249 src/lib/serif/geometry/deformation/determinant_polynomial.cpp File Reference . . . 1379
11.250 src/lib/serif/geometry/deformation/harmonics.cpp File Reference . . . 1380
11.251 src/lib/serif/geometry/deformation/radial_extension.cpp File Reference . . . 1381
11.252 src/lib/serif/geometry/deformation/surface_embedding.cpp File Reference . . . 1381
11.253 src/lib/serif/geometry/surface/isobaric.cpp File Reference . . . 1382
11.254 src/lib/serif/mapping/coefficients.cpp File Reference . . . 1383
11.255 src/lib/serif/mapping/compactification/kelvin.cpp File Reference . . . 1383
11.256 src/lib/serif/mapping/data.cpp File Reference . . . 1384
11.257 src/lib/serif/mapping/domain_mapper.cpp File Reference . . . 1384
11.258 src/lib/serif/mapping/evaluator.cpp File Reference . . . 1385
11.259 src/lib/serif/mapping/transformations.cpp File Reference . . . 1385

11.260 src/lib/serif/mapping/volume_cache.cpp File Reference 1387

Index 1389

Chapter 1

SERiF

This is an initial implementation of the numerics of SERiF

Chapter 2

Todo List

File dependencies.hpp

We may want to remove the ability to have implicit casting between types, this is a bit of a foot-gun and may lead to subtle bugs.

We really should remove the try_prepare'dness of it all, that function uses exceptions for runtime logic which would make Herb Sutter very mad (and on a more serious note can lead to significant performance issues as the stack is unwound). This is a larger refactor as we need to propegate value based error handeling all through the code but should be done at some point.

File forms.hpp

Come back and fill in more information about elsewhere. If you are the developer from above and you are reading this todo, then I am, and I mean this truly, very sorry....

Struct serif::discretization::backend::mfem::TensorBasisCache::Entry

Consider if we can remove the shared pointer from here, shared pointers introduce overhead with reference counting and that may slow down the cache.

Member serif::discretization::blocks::FunctionalSpaceOf

For consistency with other types consider renaming TermT::FunctionalSpace to TermT::functional_space. This would require updating all term definitions to use the new naming convention. CLion or some other IDE should be able to handle this refactor easily.

Struct serif::discretization::blocks::generated_residual_block< GeneratedResidual >

This may be removable at this point

Struct serif::discretization::blocks::generated_value_block< GeneratedValue >

This may be removable at this point

Member serif::discretization::blocks::QuantityOf

For consistency with other types consider renaming TermT::Quantity to TermT::quantity. This would require updating all term definitions to use the new naming convention. CLion or some other IDE should be able to handle this refactor easily.

Member serif::discretization::blocks::SupportOf

For consistency with other types consider renaming TermT::Support to TermT::support. This would require updating all term definitions to use the new naming convention. CLion or some other IDE should be able to handle this refactor easily.

Member serif::discretization::blocks::TermsOf

For consistency with other types consider renaming FieldT::Terms to FieldT::terms. This would require updating all field definitions to use the new naming convention. CLion or some other IDE should be able to handle this refactor easily.

Member serif::discretization::domain::ids::validation::all_unique_values (Ts... values) noexcept

Really this should probably be refactored into the utils module since this is not super specific to domainIDs.

Member serif::discretization::forms::component_count

This introduces an assumption of three-dimensionality, we say "this is either a scalar or a tensor of dimension 3". We should probably relax this assumption.

Member serif::discretization::forms::PartialDerivative

Should probably be consolidated with the other PartialDerivative system... No, should definitely be consolidated...

Member serif::discretization::forms::VectorValue

This file was a mistake, the information it is is helpful yes, however, it has lost generality, lost good type information, and does not need to be separate Really we should refactor this to be part of Operation, remove the assumption of 3 dimensionality, and probably fall back to something like a span rather than an array.

Member serif::discretization::functional_space::Continuity

There may be a better way to store these relations that forces them to be parallel rather than just relying on the programmer to keep them in sync.

Member serif::operators::backend::mfem::TensorBasisCache::get (const ::mfem::FiniteElement &element, const ::mfem::IntegrationRule &rule, ReferenceOperation operation, const int copies)

I think it would make sense, at some point, to switch this to an std::optional return

File value.hpp

This should be consolidated and removed. This information could be injected into the Operation itself.

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

mfem . . . 57
serif . . . 57
serif::dimensions . . . 57
serif::dimensions::runtime . . . 67
serif::discretization . . . 68
serif::discretization::backend . . . 68
serif::discretization::backend::mfem . . . 69
serif::discretization::blocks . . . 72
serif::discretization::domain . . . 77
serif::discretization::domain::ids . . . 79
serif::discretization::domain::ids::lists . . . 81
serif::discretization::domain::ids::validation . . . 83
serif::discretization::domain::mesh . . . 84
serif::discretization::domain::relation . . . 84
serif::discretization::domain::relation::lists . . . 85
serif::discretization::domain::relation::validation . . . 86
serif::discretization::domain::schema . . . 87
serif::discretization::domain::schema::validation . . . 90
serif::discretization::forms . . . 91
serif::discretization::functional_space . . . 96
serif::discretization::quadrature . . . 102
serif::discretization::quadrature::backend . . . 104
serif::discretization::quadrature::backend::mfem . . . 105
serif::eos . . . 106
serif::eos::models . . . 107
serif::eos::relations . . . 110
serif::eos::runtime . . . 113
serif::equations . . . 116
serif::equations::gravity_contributions . . . 117
serif::geometry . . . 118
serif::geometry::deformation . . . 119

serif::geometry::deformation::backend . . . 126
serif::geometry::deformation::backend::mfem . . . 126
serif::geometry::deformation::backend::mfem::detail . . . 132
serif::geometry::deformation::backend::mfem::volume_displacement_keys . . . 134
serif::geometry::deformation::detail . . . 134
serif::geometry::deformation::keys . . . 137
serif::geometry::deformation::spaces . . . 137
serif::geometry::surface . . . 137
serif::geometry::surface::detail . . . 141
serif::geometry::surface::keys . . . 148
serif::mapping . . . 148
serif::mapping::compactification . . . 153
serif::meta . . . 154
serif::operators . . . 155
serif::operators::backend . . . 159
serif::operators::backend::mfem . . . 159
serif::operators::backend::mfem::nodal_keys . . . 192
serif::operators::backend::mfem::volume_keys . . . 192
serif::operators::detail . . . 192
serif::utils . . . 193
serif::utils::checks . . . 193
serif::utils::dependencies . . . 194
serif::utils::dependencies::detail . . . 204
serif::utils::dependencies::keys . . . 207
serif::utils::errors . . . 207
serif::utils::misc . . . 211
serif::utils::misc::concepts . . . 212
serif::utils::misc::format . . . 213
serif::utils::misc::std_helper . . . 214
serif::utils::misc::terminal . . . 215
serif::utils::misc::traits . . . 219
serif::utils::types . . . 219
serif::utils::types::backend . . . 219
serif::utils::types::backend::mfem . . . 220
serif::utils::user_options . . . 220
serif::utils::warnings . . . 220

Chapter 4

Concept Index

4.1 Concepts

Here is a list of all concepts with brief descriptions:

@1200271712111733323463461531470020022660652233266::HasBoundType . . . 221
@1200271712111733323463461531470020022660652233266::HasBoundTypeMember . . . 221
@2000743173031670560630642560700462030540640522214::CanInspectChange . . . 221
@2000743173031670560630642560700462030540640522214::CanMakeCache . . . 222
@2000743173031670560630642560700462030540640522214::CanMakeComputation . . . 222
@2000743173031670560630642560700462030540640522214::CanMakeFallibleCache . . . 222
@2000743173031670560630642560700462030540640522214::CanRequest . . . 222
@2303720751251171121462160220502573311771371073327::CanCompose
Equal vector lengths cannot substitute for matching physical meanings at a composition boundary . . . 222
@2470062770772112161533503411423573371610010261433::HasSurfaceRelation . . . 223
@2470062770772112161533503411423573371610010261433::SelectsCarrier . . . 223
@2521661430352643140212142460260550411201633220413::CanResolve . . . 223
@3231723273641022352103333370570650151523543313433::CanBorrow . . . 223
@3231723273641022352103333370570650151523543313433::CanMakeContext . . . 224
@3231723273641022352103333370570650151523543313433::CanRead . . . 224
@3231723273641022352103333370570650151523543313433::CanSelect . . . 224
@3471452410341021410720043571712743602303460643563::CanMakeStore . . . 224
@3471452410341021410720043571712743602303460643563::CanOwn . . . 224
@3471452410341021410720043571712743602303460643563::CanRead . . . 225
@3471452410341021410720043571712743602303460643563::CanReadEntry . . . 225
serif::dimensions::PhysicalQuantityType
Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity. Note all ThermodynamicQuantities will also satisfy this concept . . . 225
serif::dimensions::QuantityValueType
Turn those compile time structs into a concept that can be used to constrain a template parameter . . . 226
serif::dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity . . . 226

serif::dimensions::ThermodynamicQuantityType
Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity (both enforced through checking PhysicalQuantity↩ Type) and then that is is also derived from ThermodynamicQuantity. Note not all physical quanties will satisfy this . . . 226
serif::dimensions::ThermodynamicQuantityValueType . . . 227
serif::discretization::blocks::Field
Concept for field types in the discretization framework. To be a field a type must derive from the field base class . . . 227
serif::discretization::blocks::GlobalScalarTerm
Concept for term types that are global scalar terms. A global scalar term is a term that has support (HasSupport) but does not have a functional space (HasFunctionalSpace) and is derived from the global_scalar class . . . 228
serif::discretization::blocks::HasFunctionalSpace
Concept for term types that have a functional space. To have a functional space a type must first be a valid Term and must have a FunctionalSpace member type . . . 228
serif::discretization::blocks::HasQuantity
Concept for term types that have a physical quantity associated with them. To have a quantity a type must first be a valid Term and must have a Quantity member type . . . 228
serif::discretization::blocks::HasResidualBlock
Concept for term types that have a residual block associated with them. To have a residual block a type must first be a valid Term and must have a residual member type . . . 229
serif::discretization::blocks::HasSupport
Concept for term types that have a domain over which they are supported. To have support a type must first be a valid Term and must have a Support member type . . . 229
serif::discretization::blocks::HasValueBlock
Concept for term types that have a value block associated with them. To have a value block a type must first be a valid Term and must have a value member type . . . 230
serif::discretization::blocks::PhysicalSpatialTerm
Concept for a physical spatial term. A physical spatial term is a spatial term (SpatialTerm) that has a physical quantity associated with it (HasQuantity) . . . 230
serif::discretization::blocks::SpatialTerm
Concept for term types that are spatial terms. A spatial term is a term that has a functional space (HasFunctionalSpace) and a support (HasSupport) . . . 231
serif::discretization::blocks::Term
Concept for term types in the discretization framework. To be a term a type must derive from term_base, must have a field_type member, and that field_type member must be a valid field . . . 231
serif::discretization::domain::BoundaryIsIdentifiedBy . . . 231
serif::discretization::domain::DomainIsIdentifiedBy . . . 232
serif::discretization::domain::ids::IsBoundaryID
Concept for types that are boundary IDs in the discretization framework. To be a boundary ID a type must satisfy the is_boundary_id_v trait . . . 232
serif::discretization::domain::ids::IsDomainID
Concept for types that are domain IDs in the discretization framework. To be a domain ID a type must satisfy the is_domain_id_v trait . . . 232
serif::discretization::domain::ids::IsDomainOrBoundaryID
Concept for types that are either domain IDs or boundary IDs in the discretization framework . . . 233
serif::discretization::domain::ids::lists::IsBoundaryIDList
Concept for types that are boundary ID lists in the discretization framework. To be a boundary ID list a type must satisfy the is_boundary_id_list_v trait . . . 233

serif::discretization::domain::ids::lists::IsDomainIDList
Concept for types that are domain ID lists in the discretization framework. To be a domain ID list a type must satisfy the is_domain_id_list_v trait . . . 234
serif::discretization::domain::ids::validation::AreDistinctBoundaryIDs
Concept to check if a pack of boundary IDs are distinct. This means that each ID is a valid boundary ID, no two IDs have the same integer value, and no two IDs refer to the same boundary type . . . 234
serif::discretization::domain::ids::validation::AreDistinctDomainIDs
Concept to check if a pack of domain IDs are distinct. This means that each ID is a valid domain ID, no two IDs have the same integer value, and no two IDs refer to the same domain type . . . 235
serif::discretization::domain::ids::validation::HaveUniqueBoundaries
Concept to check if a pack of boundary IDs have unique boundary types. This is used to ensure that no two boundary IDs in a list refer to the same boundary type . . . 235
serif::discretization::domain::ids::validation::HaveUniqueBoundaryIDs
Concept to check if a pack of boundary IDs have unique ID values. This is used to ensure that no two boundary IDs in a list have the same ID . . . 236
serif::discretization::domain::ids::validation::HaveUniqueDomainIDs
Concept to check if a pack of domain IDs have unique ID values. This is used to ensure that no two domain IDs in a list have the same ID . . . 236
serif::discretization::domain::ids::validation::HaveUniqueDomains
Concept to check if a pack of domain IDs have unique domain types. This is used to ensure that no two domain IDs in a list refer to the same domain type . . . 237
serif::discretization::domain::IsBoundary . . . 237
serif::discretization::domain::IsDomain . . . 237
serif::discretization::domain::IsDomainOrSet . . . 238
serif::discretization::domain::IsDomainSet . . . 238
serif::discretization::domain::relation::IsRelation . . . 238
serif::discretization::domain::relation::lists::IsRelationList . . . 238
serif::discretization::domain::relation::validation::HaveValidRelationEntities . . . 239
serif::discretization::domain::schema::IsSchema . . . 239
serif::discretization::forms::Form
Concept for types that are forms in the discretization framework. In order satisfy this concept a type must derive from the form base class and must define a member type called operand_types which is intended to be a type list of the trial and test terms for the form 239
serif::discretization::forms::IsOperand
Concept to validate a valid operand. An operand is valid if the following conditions are satisfied . . . 240
serif::discretization::forms::IsOperation . . . 240
serif::discretization::functional_space::IsSpaceDescriptor
Concept to validate a valid space descriptor. A space descriptor is valid if it is of type SpaceDescriptor. This is a very simple concept that just checks that the type is exactly SpaceDescriptor . . . 241
serif::discretization::quadrature::backend::mfem::ConfigurableIntegrator . . . 241
serif::discretization::quadrature::backend::mfem::HasPointerSetIntegrationRule . . . 241
serif::discretization::quadrature::backend::mfem::HasReferenceSetIntegrationRule . . . 242
serif::discretization::quadrature::backend::mfem::HasSetIntRule . . . 242
serif::eos::models::BarotropicClosureEOS . . . 242
serif::eos::models::EOSModel . . . 242
serif::eos::models::EOSSupportingPartialDerivative . . . 243
serif::eos::models::EOSSupportingRelation . . . 243
serif::eos::models::PressureForceEOS . . . 243
serif::eos::models::StructureSeedEOS . . . 243
serif::eos::relations::CatalogContainingRelation . . . 243

serif::eos::relations::RelationContainingInput . . . 244
serif::eos::relations::ValidRelationCatalog . . . 244
serif::eos::relations::ValidThermodynamicRelation . . . 244
serif::eos::runtime::RuntimeEOSModel . . . 244
serif::equations::Equation
Physics authoring contract; contributions share one residual row but may use different tests/supports . . . 245
serif::equations::PointEquation . . . 245
serif::geometry::deformation::DeformationMap . . . 245
serif::geometry::deformation::DifferentiableDeformationMap
Nonlinear maps explicitly provide every term needed by a Newton force pullback . . . 246
serif::geometry::deformation::LinearDeformationMap
Fixed maps supply their forward and work-conjugate transpose once . . . 246
serif::geometry::surface::DirectPressureSurfaceCompilable
A direct pressure residual requires an explicitly chosen pressure relation and an input quantity whose equation row it replaces . . . 247
serif::geometry::surface::IsobaricSurfaceType . . . 247
serif::geometry::surface::PressureSurfaceCompilable
Whether the available state selects exactly one pressure-to-carrier EOS relation . . . 247
serif::geometry::surface::PressureSurfaceFormulationType . . . 248
serif::geometry::surface::SurfaceResidualRow
The quantity enforcing the boundary must have an actual equation row, not just sampled values . . . 248
serif::geometry::surface::SurfaceScalarTerm
Scalar thermodynamic data may come from any declared scalar space; boundary enforcement is a separate backend capability . . . 248
serif::geometry::surface::SurfaceStateBindings
Reuse dependency bindings, retaining distinct quantity roles even when they refer to one term . . . 249
serif::operators::backend::mfem::AffinePointEquation
An affine scalar point condition supplies the two off-diagonal border blocks . . . 249
serif::operators::backend::mfem::ClosureInputsAccept . . . 250
serif::operators::backend::mfem::ClosureResult . . . 250
serif::operators::backend::mfem::EvaluableOperand
Check the complete interface used by preparation, Jacobian application, and its transpose 250
serif::operators::CanSubtract . . . 251
serif::operators::CanSum . . . 251
serif::operators::NodalValueEquation
Nodal execution: an H1 value depending only on itself, with no explicit position dependence 251
serif::operators::ScalarValueOperand . . . 252
serif::operators::ScalarValueTerm . . . 252
serif::operators::ScalarVolumeEquation
Equations executable by the current scalar-value volume operator . . . 252
serif::operators::VolumeEquation
One or more supported weak volume contributions assembled into a single residual row . 253
serif::operators::VolumeOperand
Mathematical operand combinations implemented by the current volume backend . . . 253
serif::utils::dependencies::BindingsWithBinding . . . 253
serif::utils::dependencies::CanPrepare . . . 254
serif::utils::dependencies::CompatibleComputations . . . 254
serif::utils::dependencies::ComputationGraphFor . . . 254
serif::utils::dependencies::IsBinding . . . 254
serif::utils::dependencies::IsBindings . . . 255
serif::utils::dependencies::IsComputation . . . 255

serif::utils::dependencies::IsComputationGraph 255
serif::utils::dependencies::IsDependencyContext 255
serif::utils::dependencies::IsKeyList 256
serif::utils::dependencies::IsOwnedComputation 256
serif::utils::dependencies::IsResource 256
serif::utils::dependencies::SatisfiesKeyList 256
serif::utils::dependencies::SnapshotPolicyFor 256
serif::utils::misc::concepts::AllTypesAreUnique 257
serif::utils::misc::concepts::IsComparableToNumeric 257
serif::utils::misc::concepts::IsEnum 257
serif::utils::misc::concepts::IsException 257
serif::utils::misc::concepts::IsExceptionWithErrorCode 258
serif::utils::misc::concepts::IsNumeric 258
serif::utils::misc::concepts::IsOrderable 258
serif::utils::misc::concepts::SetOfOneOrTwo 258

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

serif::utils::dependencies::detail::AppendBinding< BindingsT, BindingT > . . . 270
serif::utils::dependencies::detail::AppendBinding< Bindings< BindingsTs... >, BindingT > . . . 270
serif::utils::dependencies::detail::AppendKeys< ResultT, Keys > . . . 271
serif::utils::dependencies::detail::AppendKeys< AppendUnique< ResultT, FirstT >::type, Rest... > 271
serif::utils::dependencies::detail::AppendKeys< ResultT, FirstT, Rest... > . . . 271
serif::utils::dependencies::detail::AppendUnique< ListT, T > . . . 272
serif::utils::dependencies::detail::AppendUnique< KeyList< Keys... >, T > . . . 272
serif::operators::detail::AsKeyList< ListT > . . . 276
serif::operators::detail::AsKeyList< utils::types::TypeList< T... > > . . . 277
serif::operators::BadContributionCollection . . . 277
serif::operators::BadContributionRow . . . 279
serif::equations::BarotropicClosure< EOST > . . . 282
serif::equations::BarotropicHydrostaticEquilibrium . . . 285
serif::operators::BarotropicHydrostaticEquilibrium . . . 287
serif::operators::backend::mfem::volume_keys::Basis< Contribution > . . . 290
serif::operators::backend::mfem::BasisBindings< ColumnCount > . . . 290
serif::discretization::backend::mfem::BasisEvaluation . . . 292
serif::operators::backend::mfem::BasisEvaluation . . . 300
serif::discretization::backend::mfem::BasisWorkspace . . . 307
serif::operators::backend::mfem::BasisWorkspace . . . 308
serif::utils::dependencies::Binding< KeyT, TargetT > . . . 309
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< K, T > . . . 309
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< BindingTs::Key, BindingTs::Target > . . . 309
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::collector . . . 354
serif::utils::dependencies::Bindings< BindingTs > . . . 310
serif::discretization::blocks::block . . . 310
serif::discretization::blocks::residual_block_base . . . 868
serif::discretization::blocks::density::mass::residual . . . 854

serif::discretization::blocks::displacement::vector::residual . . . 856
serif::discretization::blocks::enthalpy::specific::residual . . . 857
serif::discretization::blocks::generated_residual_block< GeneratedResidual > . . . 587
serif::discretization::blocks::global_scalar::residual . . . 859
serif::discretization::blocks::gravity::gradient::residual . . . 861
serif::discretization::blocks::gravity::potential::residual . . . 862
serif::discretization::blocks::residual_block< index_value > . . . 865
serif::discretization::blocks::surface_deformation::shape_equilibrium::residual . . . 864
serif::discretization::blocks::value_block_base . . . 996
serif::discretization::blocks::density::mass::value . . . 983
serif::discretization::blocks::displacement::vector::value . . . 984
serif::discretization::blocks::enthalpy::specific::value . . . 986
serif::discretization::blocks::generated_value_block< GeneratedValue > . . . 589
serif::discretization::blocks::global_scalar::value . . . 987
serif::discretization::blocks::gravity::gradient::value . . . 989
serif::discretization::blocks::gravity::potential::value . . . 991
serif::discretization::blocks::surface_deformation::coefficients::value . . . 992
serif::discretization::blocks::value_block< index_value > . . . 994
std::bool_constant
serif::discretization::blocks::IsGlobalScalarTerm< T > . . . 644
serif::discretization::blocks::IsPhysicalSpatialTerm< T > . . . 648
serif::discretization::blocks::IsSpatialTerm< T > . . . 652
serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< Domain↩ Boundary< BoundaryT, DomainTs... >, DomainIDsT, BoundaryIDsT > . . . 847
serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< Fully↩ Connected< DomainT >, DomainIDsT, BoundaryIDsT > . . . 848
serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< Inscribed< InnerDomainT, OuterDomainT >, DomainIDsT, BoundaryIDsT > . . . 849
serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< Domain↩ IDsT, BoundaryIDsT, lists::RelationList< RelationTs... > > . . . 846
serif::eos::models::ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > > . . . 628
serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std↩ ::remove_cvref_t< EOSCandidate >::Relations > > . . . 643
serif::eos::relations::CatalogContainsRelation< RelationCatalog< Relations... >, RelationT > . . . 334
serif::eos::relations::IsThermodynamicRelation< Relation< Output, Inputs... > > . . . 654
serif::eos::relations::RelationContainsInput< Relation< Output, Inputs... >, Quantity > . . . 843
serif::eos::runtime::HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > > . . . 622
serif::eos::runtime::RuntimeRelationsAreSupported< true, RelationTs... > . . . 903
serif::equations::ValidContributions< RowT, std::tuple< C... > > . . . 982
serif::geometry::surface::detail::AreSurfaceBindings< utils::dependencies::Bindings< Entries... > > . . . 276
serif::operators::VolumeContributions< std::tuple< C... > > . . . 1009
serif::utils::dependencies::KeyListBindings< BindingsT, KeyList< KeyTs... > > . . . 666
serif::utils::dependencies::detail::UsedResource< GraphT >::Predicate< ResourceT > . . . 778
std::bool_constant< sizeof...(O) !=0 &&(VolumeOperand< O > &&...) &&(requires(const C &contribution, O operand) { {contribution.additional_order(operand, 1)}->std::same_as< int >;} &&...)>
serif::operators::VolumeOperands< C, utils::types::TypeList< O... > > . . . 1020
std::bool_constant< sizeof...(Operands) !=0 &&(ScalarValueOperand< Operands > &&...) &&(requires(const EquationT &equation, Operands operand) { {equation.additional_order(operand, 1)}->std↩ ::same_as< int >;} &&...)>
serif::operators::HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > > . . . 624

std::bool_constant<(sizeof...(Relations) > 0) &&(IsThermodynamicRelation< Relations >::value &&...) &&AllTypesAreUnique< Relations... > >
serif::eos::relations::IsValidRelationCatalog< RelationCatalog< Relations... > > . . . 656
serif::operators::backend::mfem::nodal_keys::BorderLinearization . . . 314
serif::utils::dependencies::detail::BorrowedValue< T > . . . 314
serif::discretization::domain::Boundary . . . 315
serif::discretization::domain::InfinitySurfaceBoundary . . . 631
serif::discretization::domain::StellarSurfaceBoundary . . . 935
serif::discretization::domain::ids::BoundaryDescriptor . . . 318
serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue > . . . 319
serif::discretization::domain::schema::BoundaryID< BoundaryT, IdentifierValue > . . . 321
serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs > . . . 322
serif::discretization::domain::schema::BoundaryIDList< BoundaryIDTs > . . . 323
serif::discretization::domain::BoundaryIDResolver< BoundaryT, BoundaryIDListT > . . . 324
serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > > . . . 325
serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > > . . . 326
serif::discretization::domain::BoundarySet< BoundaryTs > . . . 327
serif::geometry::surface::BoundaryValues< Term > . . . 327
serif::utils::dependencies::detail::BoundType< BindingsT, KeyT > . . . 331
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT > . . . 331
serif::utils::dependencies::detail::BoundType< std::remove_cvref_t< BindingsT >, KeyT > . . . 331
serif::utils::dependencies::target_type< BindingsT, KeyT > . . . 330
serif::utils::dependencies::BoundType< BindingsT, KeyT > . . . 330
serif::utils::dependencies::detail::CachedResult< ValueT, InputCount > . . . 332
serif::geometry::deformation::backend::mfem::volume_displacement_keys::CheckedBinding . . . 343
mfem::Coefficient
serif::mapping::PhysicalPositionFunctionCoefficient . . . 756
serif::geometry::deformation::backend::mfem::volume_displacement_keys::Coefficients . . . 345
serif::discretization::backend::mfem::CoefficientTracking . . . 347
serif::operators::backend::mfem::NodalValueTracking . . . 720
serif::operators::backend::mfem::CoefficientTracking . . . 350
serif::operators::backend::mfem::NodalValueTracking . . . 720
serif::operators::backend::mfem::CoefficientView< BindingT > . . . 353
serif::operators::detail::Columns< ListT > . . . 355
serif::operators::detail::Columns< utils::types::TypeList< O... > > . . . 355
serif::operators::backend::mfem::CompactificationCoefficients . . . 356
serif::mapping::CompactificationPointData . . . 356
serif::utils::CompensatedSum . . . 356
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap > . . . 358
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT > . . . 363
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT > . . . 365
serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, ResourcesT, IndicesT > . . . 372
serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > > . . . 373
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, Valid > . . . 374
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true > . . . 374
serif::utils::dependencies::ComputationGraph< Computations > . . . 376
serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, Independent > . . . 378
serif::utils::dependencies::ComputationStore< EntryTs > . . . 380
serif::utils::dependencies::ComputationUpdate< InputsT, ValueT > . . . 383

serif::utils::misc::concepts::Concat< Lists > . . . 384
serif::utils::misc::concepts::Concat< types::TypeList< LhsTs... >, types::TypeList< RhsTs... >, Rest... > . . . 384
serif::utils::misc::concepts::Concat< types::TypeList< Ts... > > . . . 385
serif::utils::misc::concepts::Concat<> . . . 385
serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics . . . 386
serif::discretization::domain::schema::validation::RelationValidationResult::ConnectedDiagnostics . . . 386
serif::equations::Contribution< TestOperandT, SupportT > . . . 387
serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::barotropic_constant::mass_normalization > > . . . 387
serif::equations::MassNormalization . . . 693
serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::displacement::vector > > . . . 387
serif::equations::CenteringConstraint . . . 335
serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::displacement::vector >, discretization::domain::StellarDomains > . . . 387
serif::equations::GravityForce . . . 611
serif::equations::RotationalForce . . . 888
serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::displacement::vector, discretization::forms::Operation::Divergence >, discretization::domain::StellarDomains > . . . 387
serif::equations::PressureForce< eos::models::Polytrope > . . . 787
serif::equations::PressureForce< EOST > . . . 787
serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::enthalpy::specific > > . . . 387
serif::equations::CentralDensityConstraint< eos::models::Polytrope > . . . 339
serif::equations::CentralDensityConstraint< EOST > . . . 339
serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::enthalpy::specific >, discretization::domain::StellarSurfaceBoundary > . . . 387
serif::equations::PressureSurfaceConstraint< eos::models::Polytrope > . . . 790
serif::equations::PressureSurfaceConstraint< EOST > . . . 790
serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::rotation::angular_velocity > > . . . 387
serif::equations::AngularMomentum . . . 265
serif::equations::Contribution< discretization::forms::Operand< TermT, discretization::forms::Operation::Value >, discretization::domain::StellarSurfaceBoundary > . . . 387
serif::operators::NodalRequest< TermT, OperationT, PositionIndependent > . . . 718
serif::equations::Contribution< Operand< Gradient > > . . . 387
serif::equations::gravity_contributions::GradientValue . . . 603
serif::equations::Contribution< Operand< Gradient, Divergence > > . . . 387
serif::equations::gravity_contributions::PotentialValue . . . 774
serif::equations::Contribution< Operand< Potential > > . . . 387
serif::equations::gravity_contributions::GradientDivergence . . . 599
serif::equations::Contribution< Operand< Potential >, discretization::domain::StellarDomains > . . . 387
serif::equations::gravity_contributions::DensitySource . . . 411
serif::operators::ContributionBinding< EquationT > . . . 388
serif::operators::detail::ContributionChildren< EquationT > . . . 389
serif::operators::backend::mfem::ContributionData< OperandCount > . . . 391
serif::operators::detail::ContributionOperands< TupleT > . . . 391
serif::equations::Contributions< T, typename > . . . 393
serif::equations::ContributionSupport< T, typename > . . . 394

serif::equations::ContributionSupport< T, std::void_t< typename T::Support > > . . . 395
serif::equations::ContributionTuple< TupleT, typename > . . . 395
serif::equations::ContributionTuple< std::remove_cvref_t< decltype(std::declval< const T & >().contributions())> > . . . 395
serif::equations::Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions())> > . . . 393
serif::equations::ContributionTuple< std::tuple< Ts... >, std::void_t< decltype(std::tuple_cat(std::declval< typename Contributions< std::remove_cvref_t< Ts > >::type >()...))> > . . . 396
serif::utils::dependencies::keys::Coordinates . . . 397
serif::operators::detail::CouplingForm< Input, TestOperand > . . . 398
serif::utils::types::ct_error_string< N > . . . 399
serif::utils::types::ctstring< N > . . . 400
serif::discretization::forms::Operation::Curl . . . 401
serif::geometry::deformation::DeformationFailure . . . 402
serif::geometry::deformation::DeformationLinearization< Map > . . . 403
serif::geometry::deformation::DeformationWorkspaceType< Map, Linear > . . . 406
serif::geometry::deformation::DeformationWorkspaceType< Map, true > . . . 407
serif::utils::types::DenseMatrixBase< T > . . . 407
serif::utils::types::DenseTensorBase< T, Rank > . . . 408
serif::utils::dependencies::DependencyContext< ResourceTs > . . . 413
serif::utils::dependencies::DependencyContextView< ParentT, KeysT > . . . 415
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > > . . . 415
serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > . . . 421
serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< FunctionalSpaceT > . . . 422
serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > > . . . 423
serif::operators::backend::mfem::DisplacementCoefficients . . . 425
serif::equations::DisplacementForce< EOST > . . . 426
serif::discretization::forms::Operation::Divergence . . . 428
serif::mapping::DMFixedMeshGeometry . . . 431
serif::mapping::DMGeometryCache . . . 431
serif::mapping::DMGeometryCacheStatistics . . . 433
serif::mapping::DMReferenceDisplacementSample . . . 433
serif::mapping::DMWorkspace . . . 434
serif::discretization::backend::mfem::DofMap . . . 437
serif::discretization::backend::mfem::TermBinding< Closure::Test, Schema > . . . 964
serif::discretization::backend::mfem::TermBinding< discretization::blocks::enthalpy::specific, Schema > . . . 964
serif::discretization::backend::mfem::TermBinding< Centering::Test, Schema > . . . 964
serif::discretization::backend::mfem::TermBinding< Test, SchemaT > . . . 964
serif::discretization::backend::mfem::TermBinding< typename EquationT::Border, SchemaT > . . . 964
serif::discretization::backend::mfem::TermBinding< TestT, SchemaT > . . . 964
serif::discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, SchemaT > . . . 964
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT > . . . 964
serif::discretization::backend::mfem::DofSelection . . . 445
serif::discretization::domain::Domain . . . 449
serif::discretization::domain::CoreDomain . . . 397
serif::discretization::domain::EnvelopeDomain . . . 503
serif::discretization::domain::VacuumDomain . . . 979
serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics . . . 454

serif::discretization::domain::schema::validation::RelationValidationResult::DomainBoundary↩
Diagnostics . . . 455
serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT > . . . 457
serif::discretization::domain::ids::DomainDescriptor . . . 458
serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue > . . . 460
serif::discretization::domain::schema::DomainID< DomainT, IdentifierValue > . . . 461
serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs > . . . 462
serif::discretization::domain::schema::DomainIDList< DomainIDTs > . . . 463
serif::discretization::domain::DomainIDResolver< DomainT, DomainIDListT > . . . 465
serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::Domain↩
IDList< DomainIDTs... > > . . . 465
serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< Domain↩
IDTs... > > . . . 466
serif::geometry::deformation::keys::DomainMap . . . 467
serif::mapping::DomainMapper . . . 467
serif::utils::user_options::DomainMapperOptions . . . 474
serif::discretization::domain::DomainOrSet . . . 475
serif::discretization::domain::DomainOrSetList< DomainTs > . . . 476
serif::discretization::domain::relation::DomainRelation . . . 477
serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs > . . . 450
serif::discretization::domain::relation::FullyConnected< A > . . . 584
serif::discretization::domain::relation::Inscribed< A, B > . . . 633
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations > . . . 477
serif::discretization::domain::DomainSet< DomainTs > . . . 481
serif::operators::backend::mfem::ElementBasis< OperandCount > . . . 484
serif::mapping::ElementCompactificationData . . . 485
serif::operators::backend::mfem::ElementData< OperandCount > . . . 486
serif::mapping::ElementDisplacementData . . . 488
serif::operators::backend::mfem::ElementGeometry< OperandCount > . . . 490
serif::mapping::ElementMappingData . . . 491
serif::geometry::deformation::EmptyDeformationWorkspace . . . 492
serif::discretization::backend::mfem::TensorBasisCache::Entry . . . 499
serif::operators::backend::mfem::TensorBasisCache::Entry . . . 501
serif::utils::misc::concepts::EnumerateTypes< TypeList, Descriptor > . . . 502
serif::utils::misc::concepts::EnumerateTypes< types::TypeList< TypeTs... >, Descriptor > . . . 502
serif::geometry::surface::keys::EOSResponse . . . 505
serif::eos::runtime::EOSView . . . 506
serif::equations::EquationDifference< LeftT, RightT > . . . 510
serif::geometry::surface::keys::EquationOfState . . . 512
serif::operators::EquationPlan< EquationT, OperandsT > . . . 512
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > > . . . 512
serif::equations::EquationPoint< OperandsT > . . . 515
serif::equations::EquationRegistry< EquationTs > . . . 516
serif::operators::EquationRegistry< EquationTs > . . . 518
serif::equations::EquationScale< EquationT > . . . 520
serif::equations::EquationSum< FirstT, RestTs > . . . 522
std::exception
std::logic_error
std::domain_error
serif::eos::EOSEvaluationError . . . 504
serif::mapping::compactification::ExteriorDomainMap . . . 524
serif::mapping::compactification::KelvinCompactification . . . 661
serif::mapping::compactification::ExteriorMapDirection . . . 526
serif::mapping::compactification::ExteriorMapInput . . . 526

serif::mapping::compactification::ExteriorMapResult . . . 527
serif::mapping::compactification::ExteriorMapVariation . . . 528
serif::discretization::domain::mesh::FaceElements . . . 529
serif::mapping::FaceMappingContext . . . 529
serif::mapping::FaceMappingVariation . . . 531
serif::mapping::FaceQuadratureContext . . . 532
std::false_type
serif::dimensions::IsQuantityValue< QuantityValue< Quantity > > . . . 651
serif::eos::models::ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > > . . . 628
serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std::remove_cvref_t< EOSCandidate >::Relations > > . . . 643
serif::eos::relations::CatalogContainsRelation< RelationCatalog< Relations... >, RelationT > . . . 334
serif::eos::relations::IsThermodynamicRelation< Relation< Output, Inputs... > > . . . 654
serif::eos::relations::IsValidRelationCatalog< RelationCatalog< Relations... > > . . . 656
serif::eos::relations::RelationContainsInput< Relation< Output, Inputs... >, Quantity > . . . 843
serif::eos::runtime::HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > > . . . 622
serif::eos::runtime::RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > > 897
serif::eos::runtime::RuntimeRelationsAreSupported<(HasRuntimeQuantityIdentifiers< RelationTs >::value &&...), RelationTs... > . . . 902
serif::eos::runtime::RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > > . . . 897
serif::eos::runtime::RuntimeRelationsAreSupported< true, RelationTs... > . . . 903
serif::equations::ValidContributions< RowT, std::tuple< C... > > . . . 982
serif::geometry::surface::detail::AreSurfaceBindings< utils::dependencies::Bindings< Entries... > > . . . 276
serif::operators::HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > > . . . 624
serif::operators::VolumeContributions< std::tuple< C... > > . . . 1009
serif::operators::VolumeOperands< C, utils::types::TypeList< O... > > . . . 1020
serif::dimensions::IsQuantityValue< Candidate > . . . 650
serif::eos::models::ImplementsPartialDerivative_t< EOS, RelationT, InputQuantity > . . . 625
serif::eos::models::ImplementsRelationCatalog_t< EOS, Output, Inputs > . . . 627
serif::eos::models::ImplementsRelation_t< EOS, RelationT > . . . 626
serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, typename > . . . 642
serif::eos::relations::CatalogContainsRelation< Catalog, RelationT > . . . 333
serif::eos::relations::IsThermodynamicRelation< RelationCandidate > . . . 653
serif::eos::relations::IsValidRelationCatalog< CandidateCatalog > . . . 655
serif::eos::relations::RelationContainsInput< RelationT, Quantity > . . . 842
serif::eos::runtime::HasRuntimeQuantityIdentifiers< RelationT > . . . 621
serif::eos::runtime::RuntimeCatalogIsSupported< CatalogT > . . . 896
serif::eos::runtime::RuntimeRelationsAreSupported< QuantitiesAreIdentified, RelationTs > . . . 902
serif::equations::ValidContributions< RowT, TupleT > . . . 981
serif::geometry::surface::detail::AreSurfaceBindings< Bindings > . . . 275
serif::operators::HasScalarVolumeOperands< EquationT, OperandsT > . . . 623
serif::operators::VolumeContributions< TupleT > . . . 1008
serif::operators::VolumeOperands< C, ListT > . . . 1019
serif::discretization::blocks::field . . . 533
serif::discretization::blocks::barotropic_constant . . . 280
serif::discretization::blocks::central_density . . . 337
serif::discretization::blocks::density . . . 409
serif::discretization::blocks::displacement . . . 424
serif::discretization::blocks::enthalpy . . . 496

serif::discretization::blocks::gravity . . . 609
serif::discretization::blocks::rotation . . . 886
serif::discretization::blocks::surface_deformation . . . 936
serif::discretization::blocks::FieldDescriptor . . . 534
serif::utils::misc::concepts::Filter< List, Predicate > . . . 535
serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate > . . . 536
serif::utils::misc::concepts::Filter< types::TypeList<>, Predicate > . . . 536
serif::geometry::deformation::FixedInfinityRadialExtension . . . 537
serif::discretization::forms::form . . . 542
serif::discretization::forms::boundary_flux< TrialT, TestT > . . . 316
serif::discretization::forms::divergence< TrialT, TestT > . . . 428
serif::discretization::forms::force< TrialT, TestT > . . . 539
serif::discretization::forms::mass< TrialT, TestT > . . . 684
serif::discretization::forms::projection< TrialT, TestT > . . . 801
serif::discretization::forms::source< TrialT, TestT > . . . 915
std::formatter
std::formatter< serif::discretization::blocks::FieldDescriptor > . . . 543
std::formatter< serif::discretization::blocks::TermDescriptor > . . . 544
std::formatter< serif::discretization::domain::DomainOrSet > . . . 546
std::formatter< serif::discretization::forms::FormDescriptor > . . . 548
std::formatter< serif::discretization::functional_space::Continuity > . . . 550
std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > > . . . 551
std::formatter< serif::discretization::functional_space::Family > . . . 553
std::formatter< serif::discretization::functional_space::Mapping > . . . 554
std::formatter< serif::discretization::functional_space::Rank< R > > . . . 556
std::formatter< serif::discretization::functional_space::SpaceDescriptor > . . . 557
std::formatter< serif::discretization::functional_space::Trace > . . . 559
std::formatter< serif::discretization::quadrature::FormRule > . . . 561
std::formatter< serif::discretization::quadrature::MappingKind > . . . 563
std::formatter< serif::discretization::quadrature::Mode > . . . 564
std::formatter< serif::discretization::quadrature::QuadratureRole > . . . 565
std::formatter< serif::discretization::quadrature::Query > . . . 567
std::formatter< serif::discretization::quadrature::RuleControl > . . . 568
std::formatter< serif::discretization::quadrature::RuleSet > . . . 569
std::formatter< serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext > . . . 560
std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult > . . . 547
serif::discretization::forms::FormDescriptor . . . 571
serif::discretization::quadrature::FormRule . . . 573
serif::discretization::forms::FormTraits< FormT > . . . 574
serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > > . . . 575
serif::discretization::forms::FormTraits< divergence< TrialT, TestT > > . . . 576
serif::discretization::forms::FormTraits< force< TrialT, TestT > > . . . 577
serif::discretization::forms::FormTraits< mass< TrialT, TestT > > . . . 578
serif::discretization::forms::FormTraits< projection< TrialT, TestT > > . . . 579
serif::discretization::forms::FormTraits< source< TrialT, TestT > > . . . 581
serif::operators::backend::mfem::volume_keys::Geometry< Contribution > . . . 591
serif::geometry::deformation::backend::mfem::GeometrySampleRule . . . 591
serif::geometry::deformation::GeometryStepLimit . . . 592
serif::geometry::deformation::GeometryStepOptions . . . 594
serif::discretization::blocks::global_scalar . . . 595
serif::discretization::blocks::barotropic_constant::mass_normalization . . . 688
serif::discretization::blocks::central_density::border . . . 312
serif::discretization::blocks::rotation::angular_velocity . . . 261

serif::discretization::forms::Operation::Gradient . . . 596
serif::operators::GradientRequest . . . 601
serif::utils::dependencies::detail::Graph< Computations > . . . 605
serif::equations::GravityGradientRelation . . . 614
serif::mapping::GridFunctionMappingEvaluator . . . 615
serif::utils::dependencies::detail::HasBinding< BindingsT, KeyT > . . . 620
serif::utils::dependencies::detail::HasBinding< Bindings< BindingTs... >, KeyT > . . . 621
serif::utils::dependencies::detail::HasBinding< std::remove_cvref_t< BindingsT >, KeyT > . . . 620
serif::utils::dependencies::HasBinding< BindingsT, KeyT > . . . 620
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT > . . . 629
serif::utils::dependencies::keys::Input . . . 633
serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics . . . 637
serif::discretization::domain::schema::validation::RelationValidationResult::InscribedDiagnostics . . . 638
std::integral_constant
serif::utils::dependencies::detail::KeyListSize< KeyList< Keys... > > . . . 667
serif::geometry::deformation::keys::InteriorModel . . . 638
serif::utils::dependencies::detail::Invocation< ComputationT, ArgumentsT, typename > . . . 640
serif::utils::dependencies::detail::Invocation< ComputationT, ResolvedTargets< BindingsT, ComputationT::Inputs > > . . . 640
serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, true > . . . 871
serif::geometry::surface::IsobaricSurface . . . 645
serif::operators::detail::Join< Lists > . . . 660
serif::operators::detail::Join< C::Operands... > . . . 660
serif::operators::detail::ContributionOperands< std::tuple< C... > > . . . 392
serif::utils::user_options::KelvinCompactificationOptions . . . 665
serif::utils::dependencies::KeyList< KeyTs > . . . 665
serif::utils::dependencies::KeyListBindings< BindingsT, KeysT > . . . 666
serif::utils::dependencies::detail::KeyListSize< KeysT > . . . 667
serif::geometry::deformation::keys::Linearization . . . 671
serif::geometry::surface::keys::Linearization . . . 672
serif::operators::backend::mfem::nodal_keys::Linearization . . . 672
serif::operators::backend::mfem::volume_keys::Linearization . . . 672
serif::geometry::deformation::backend::mfem::detail::LocatedBoundaryPoint . . . 672
serif::operators::backend::mfem::MappingCoefficients . . . 676
serif::mapping::MappingPointContext . . . 676
serif::mapping::MappingPointVariation . . . 678
serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, Catalog > . . . 695
serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Relations... > > . . . 695
serif::utils::dependencies::detail::MergeKeyLists< ResultT, Lists > . . . 696
serif::utils::dependencies::detail::MergeKeyLists< AppendKeys< ResultT, Keys... >::type, Rest... > . . . 696
serif::utils::dependencies::detail::MergeKeyLists< ResultT, KeyList< Keys... >, Rest... > . . . 697
serif::operators::backend::mfem::MeshAndCoefficientBindings . . . 698
serif::discretization::domain::mesh::MeshTopology . . . 698
serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< OperandListT > . . . 703
serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< utils::types::TypeList< OperandTs... > > . . . 703
serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext . . . 704
serif::discretization::quadrature::backend::mfem::MFEMResolver . . . 706
serif::discretization::quadrature::backend::mfem::MFEMRule . . . 708
serif::operators::NestedBadCollection . . . 712
serif::operators::backend::mfem::NodalData< EquationT, SchemaT > . . . 713
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT > . . . 714

serif::geometry::deformation::NodalRadialSurface . . . 716
serif::discretization::forms::Operation::NormalTrace . . . 721
serif::operators::backend::mfem::NumericalWork . . . 723
serif::discretization::forms::Operand< TermT, OperationT > . . . 727
serif::operators::backend::mfem::OperandEvaluator< OperandT > . . . 728
serif::operators::backend::mfem::OperandEvaluator< OperandT > . . . 728
serif::discretization::forms::OperandResult< OperandT > . . . 733
serif::operators::backend::mfem::OperandSamples . . . 735
serif::equations::OperandValues< OperandsT, OutputT > . . . 736
serif::operators::OperandValues< OperandsT, OutputT > . . . 736
serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT > . . . 737
serif::discretization::forms::Operation . . . 739
serif::discretization::forms::OperationValue< Rank, OperationT > . . . 740
serif::discretization::forms::OperationValue< 0, Operation::Gradient > . . . 740
serif::discretization::forms::OperationValue< 0, Operation::Value > . . . 741
serif::discretization::forms::OperationValue< 1, Operation::Curl > . . . 741
serif::discretization::forms::OperationValue< 1, Operation::Divergence > . . . 742
serif::discretization::forms::OperationValue< 1, Operation::NormalTrace > . . . 743
serif::discretization::forms::OperationValue< 1, Operation::Value > . . . 743
serif::discretization::forms::OperationValue< blocks::SpatialTermTraits< OperandT::term_type >::rank, OperandT::operation_type > . . . 740
serif::discretization::forms::OperandResult< OperandT > . . . 734
mfem::Operator
serif::operators::backend::mfem::JacobianOperator< CacheT > . . . 657
serif::utils::dependencies::keys::Output . . . 744
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT > . . . 744
serif::utils::dependencies::detail::OwnedValue< T > . . . 748
serif::geometry::surface::IsobaricSurface::Parameters . . . 749
serif::utils::dependencies::keys::Parameters . . . 749
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity > . . . 749
serif::dimensions::PhysicalQuantity . . . 758
serif::dimensions::Acceleration . . . 259
serif::dimensions::AngularMomentum . . . 263
serif::dimensions::AngularVelocity . . . 269
serif::dimensions::Area . . . 273
serif::dimensions::Dimensionless . . . 418
serif::dimensions::DynamicViscosity . . . 482
serif::dimensions::Energy . . . 493
serif::dimensions::Force . . . 538
serif::dimensions::Frequency . . . 582
serif::dimensions::GravitationalPotential . . . 607
serif::dimensions::KinematicViscosity . . . 668
serif::dimensions::Length . . . 670
serif::dimensions::Luminosity . . . 673
serif::dimensions::MagneticFluxDensity . . . 675
serif::dimensions::Mass . . . 679
serif::dimensions::MassFlowRate . . . 691
serif::dimensions::MomentOfInertia . . . 709
serif::dimensions::Momentum . . . 710
serif::dimensions::Opacity . . . 725
serif::dimensions::Power . . . 776
serif::dimensions::SpecificEnergy . . . 926
serif::dimensions::ThermodynamicQuantity . . . 971
serif::dimensions::ChemicalPotential . . . 343

serif::dimensions::EnergyDensity . . . 494
serif::dimensions::Entropy . . . 498
serif::dimensions::InternalEnergy . . . 639
serif::dimensions::MassDensity . . . 690
serif::dimensions::NumberDensity . . . 722
serif::dimensions::Pressure . . . 784
serif::dimensions::SpecificEnthalpy . . . 928
serif::dimensions::SpecificEntropy . . . 929
serif::dimensions::SpecificInternalEnergy . . . 931
serif::dimensions::Temperature . . . 948
serif::dimensions::Time . . . 975
serif::dimensions::Torque . . . 977
serif::dimensions::Velocity . . . 1004
serif::dimensions::Volume . . . 1005
serif::operators::backend::mfem::volume_keys::PhysicalValues< Contribution, OperandT > . . . 759
serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT > . . . 759
serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT > . . . 760
serif::equations::PointLinearization< OperandsT, ResultT > . . . 762
serif::equations::PoissonBalance . . . 763
serif::discretization::quadrature::Policy . . . 765
serif::eos::models::Polytrope . . . 766
serif::eos::models::PolytropeParameters . . . 770
serif::geometry::deformation::PowerLawRadialExtension . . . 777
serif::utils::dependencies::PreparationFailure< ErrorT > . . . 779
serif::operators::backend::mfem::PreparationRejection . . . 780
serif::utils::dependencies::PreparationReport< ResultsT, SourcesT > . . . 781
serif::utils::misc::concepts::Prepend< T, List > . . . 783
serif::utils::misc::concepts::Prepend< T, types::TypeList< Ts... > > . . . 783
serif::geometry::surface::keys::PressureCondition . . . 785
serif::geometry::surface::detail::PressureConditionTracking . . . 786
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier > . . . 793
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, pressure_relation_t< EOS, Bindings, void >::OutputQuantity, pressure_relation_t< EOS, Bindings, void >, true > . . . 793
serif::geometry::surface::PressureSurfaceFormulation< EOS, Bindings, Carrier > . . . 797
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, Relation, false > . . . 793
serif::geometry::surface::DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation > . . . 419
serif::geometry::surface::PressureSurfaceDescriptor . . . 796
serif::discretization::backend::mfem::TensorBasis::Product . . . 799
serif::utils::dependencies::ProjectedSnapshot< FunctionT > . . . 800
serif::operators::backend::mfem::volume_keys::QuadratureLinearization< Contribution > . . . 804
serif::discretization::quadrature::QuadratureManifestOptions . . . 804
serif::discretization::quadrature::QuadratureOptions . . . 805
serif::discretization::quadrature::QuadratureRoleOptions . . . 809
serif::discretization::quadrature::QuadratureTermOptions . . . 810
serif::discretization::quadrature::QuadratureValidationOptions . . . 811
serif::dimensions::QuantityAt< Index, Quantities > . . . 812
serif::dimensions::QuantityAt< Index, QuantityList< Quantities... > > . . . 812
serif::dimensions::QuantityList< Quantities > . . . 812
serif::dimensions::QuantityOf< Candidate > . . . 812

serif::dimensions::QuantityOf< QuantityValue< Quantity > > . . . 813
serif::dimensions::QuantityValue< Quantity > . . . 813
serif::discretization::quadrature::Query . . . 820
serif::geometry::deformation::RadialExtension . . . 822
serif::mapping::compactification::KelvinCompactification::RadialFactors . . . 826
serif::geometry::deformation::keys::RadialInterpolation . . . 826
serif::geometry::deformation::RadialInterpolation . . . 827
serif::discretization::functional_space::Rank< R > . . . 829
serif::geometry::deformation::backend::mfem::ReferenceDiscretization . . . 830
serif::geometry::deformation::keys::ReferenceDiscretization . . . 832
serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot . . . 832
serif::geometry::deformation::backend::mfem::ReferenceDiscretizationTracking . . . 834
serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot . . . 835
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot . . . 837
serif::geometry::deformation::keys::ReferenceSurface . . . 838
serif::geometry::deformation::ReferenceSurface . . . 839
serif::operators::backend::mfem::volume_keys::ReferenceValues< Contribution, OperandT > . . . 840
serif::eos::relations::Relation< Output, Inputs > . . . 840
serif::eos::relations::RelationCatalog< Relations > . . . 841
serif::discretization::domain::relation::lists::RelationList< RelationTs > . . . 844
serif::discretization::domain::schema::RelationList< RelationTs > . . . 844
serif::geometry::surface::detail::RelationQuantities< Relation > . . . 845
serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > > 845
serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, RelationListT > . . . 846
serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< RelationT, DomainIDsT, BoundaryIDsT > . . . 847
serif::discretization::domain::relation::validation::RelationValidationResult . . . 849
serif::discretization::domain::schema::validation::RelationValidationResult . . . 851
serif::discretization::domain::schema::validation::RelationValidator< RelationT > . . . 852
serif::discretization::domain::schema::validation::RelationValidator< DomainBoundary< BoundaryT, DomainTs... > > . . . 852
serif::discretization::domain::schema::validation::RelationValidator< FullyConnected< DomainT > > 853
serif::discretization::domain::schema::validation::RelationValidator< Inscribed< InnerT, OuterT > > 854
serif::discretization::quadrature::Resolution . . . 869
serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, Available > . . . 870
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, Done > . . . 872
serif::utils::dependencies::detail::ResolveGraph< GraphT, AppendBinding< BindingsT, Binding< ComputationT::Key, ResolveComputation< ComputationT, BindingsT >::Value > >↩ ::type, Position+1 > . . . 872
serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, true > . . . 875
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true > . . . 873
serif::utils::dependencies::detail::ResolveGraph< GraphT, ContextT::Bindings, 0 > . . . 872
serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, true > . . . 379
serif::utils::dependencies::ResolveKeys< BindingsT, KeysT > . . . 874
serif::utils::dependencies::ResolveKeys< BindingsT, KeyList< KeyTs... > > . . . 874
serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, Valid > 874
serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, GraphT::template ComputationAt< GraphT::Dependencies::order[Position]>, ResolveComputation< GraphT::template ComputationAt< GraphT::Dependencies::order[Position]>, BindingsT >::valid > . . . 874
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, false > . . . 872

serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT > . . . 876
serif::utils::dependencies::detail::ResourceKeys< ResourcesT > . . . 878
serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > > . . . 878
serif::utils::dependencies::detail::Result< T > . . . 879
serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > > . . . 880
serif::utils::dependencies::detail::Result< std::remove_cv_t< decltype(std::declval< const ComputationT & >().evaluate(std::declval< const Arguments & >()...))> > . . . 879
serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< decltype(std::declval< const ComputationT & >().evaluate(std::declval< const Arguments & >()...))> > . . . 641
serif::equations::RigidRotation . . . 882
serif::discretization::quadrature::RoleControls . . . 885
serif::discretization::quadrature::RuleControl . . . 891
serif::discretization::quadrature::RuleSet . . . 892
serif::eos::runtime::RuntimeCatalogDispatch< EOS, CatalogT > . . . 894
serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > > 895
serif::eos::runtime::RuntimeCatalogStorage< EOS, CatalogT > . . . 897
serif::eos::runtime::RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > > . 898
serif::dimensions::runtime::RuntimeQuantityValue . . . 899
serif::dimensions::runtime::RuntimeRelationDescriptor . . . 900
serif::eos::runtime::RuntimeRelationQuantities< RelationT > . . . 901
serif::eos::runtime::RuntimeRelationQuantities< relations::Relation< OutputT, InputTs... > > . . 901
serif::eos::runtime::RuntimeRelationStorage< EOS, RelationT > . . . 904
serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > > 904
serif::utils::dependencies::keys::Scale . . . 906
serif::discretization::domain::schema::validation::SchemaRelationValidationResult . . . 906
serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, RelationListT > . . . 908
serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, relation::lists::RelationList< RelationTs... > > . . . 908
serif::discretization::domain::schema::validation::SchemaValidationResult . . . 908
serif::operators::backend::mfem::SelectedNodalValues . . . 910
serif::utils::errors::serif_error_printer< Message, Code > . . . 911
serif::geometry::deformation::keys::ShapeCoefficients . . . 914
serif::geometry::deformation::spaces::ShapeParameters . . . 914
serif::utils::dependencies::detail::SourceSnapshot< ResourceT > . . . 917
serif::discretization::functional_space::SpaceDescriptor . . . 918
serif::utils::types::SparseMatrixBase< T > . . . 920
serif::utils::types::SparseTensorBase< T, Rank > . . . 920
serif::discretization::blocks::SpatialTermTraits< TermT > . . . 921
serif::geometry::deformation::SphericalHarmonicMode . . . 932
serif::geometry::deformation::SphericalHarmonicSurface . . . 933
serif::geometry::deformation::keys::SurfaceDisplacement . . . 938
serif::geometry::deformation::spaces::SurfaceDisplacement . . . 938
serif::geometry::deformation::SurfaceEmbedding . . . 939
serif::geometry::surface::SurfaceEOSResponse< Formulation > . . . 944
serif::geometry::surface::SurfaceFailure . . . 945
serif::geometry::deformation::keys::SurfaceMap . . . 946
serif::geometry::deformation::keys::SurfaceMetric . . . 946
serif::geometry::deformation::keys::SurfaceModel . . . 947
serif::geometry::surface::detail::SurfaceOperands< Terms > . . . 947
serif::geometry::surface::detail::SurfaceOperands< utils::dependencies::KeyList< Terms... > > . . . 947
serif::discretization::backend::mfem::TensorBasis . . . 949
serif::discretization::backend::mfem::TensorBasisCache . . . 955

serif::operators::backend::mfem::TensorBasisCache . . . 957
serif::discretization::backend::mfem::TensorWorkspace . . . 959
serif::discretization::blocks::term_base . . . 962
serif::discretization::blocks::term< barotropic_constant > . . . 961
serif::discretization::blocks::barotropic_constant::mass_normalization . . . 688
serif::discretization::blocks::term< central_density > . . . 961
serif::discretization::blocks::central_density::border . . . 312
serif::discretization::blocks::term< density > . . . 961
serif::discretization::blocks::density::mass . . . 681
serif::discretization::blocks::term< displacement > . . . 961
serif::discretization::blocks::displacement::vector . . . 1000
serif::discretization::blocks::term< enthalpy > . . . 961
serif::discretization::blocks::enthalpy::specific . . . 923
serif::discretization::blocks::term< gravity > . . . 961
serif::discretization::blocks::gravity::gradient . . . 596
serif::discretization::blocks::gravity::potential . . . 770
serif::discretization::blocks::term< rotation > . . . 961
serif::discretization::blocks::rotation::angular_velocity . . . 261
serif::discretization::blocks::term< surface_deformation > . . . 961
serif::discretization::blocks::surface_deformation::coefficients . . . 345
serif::discretization::blocks::surface_deformation::shape_equilibrium . . . 912
serif::discretization::blocks::term< FieldT > . . . 961
serif::discretization::blocks::TermDescriptor . . . 968
serif::equations::TestOperandTraits< T, typename > . . . 970
serif::equations::TestOperandTraits< T, std::void_t< typename T::TestOperand > > . . . 970
serif::dimensions::runtime::ThermodynamicQuantityID . . . 973
std::true_type
serif::dimensions::IsQuantityValue< QuantityValue< Quantity > > . . . 651
serif::utils::types::TypeList< Ts > . . . 978
serif::geometry::surface::detail::UniqueTerms< Terms > . . . 978
serif::geometry::surface::detail::UniqueTerms< utils::types::TypeList< Terms... > > . . . 978
serif::utils::dependencies::detail::UsedResource< GraphT > . . . 979
serif::geometry::deformation::keys::VacuumModel . . . 980
serif::discretization::forms::Operation::Value . . . 982
serif::utils::dependencies::ValueSnapshot . . . 998
serif::geometry::deformation::backend::mfem::DeformationValueTracking . . . 404
serif::geometry::surface::detail::BoundaryValueTracking . . . 329
serif::operators::backend::mfem::MetadataTracking . . . 702
serif::utils::types::VectorBase< T > . . . 1003
serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT > . . . 1007
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT > . . . 1010
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT > . . . 1012
serif::geometry::deformation::keys::VolumeDisplacement . . . 1014
serif::geometry::deformation::spaces::VolumeDisplacement . . . 1014
serif::geometry::deformation::keys::VolumeExtension . . . 1014
serif::mapping::VolumeMappingCache . . . 1015
serif::mapping::VolumeMappingContext . . . 1017
serif::mapping::VolumeMappingVariation . . . 1018
serif::operators::backend::mfem::VolumeOptions . . . 1021
serif::mapping::VolumeQuadratureContext . . . 1022
serif::operators::backend::mfem::VolumeScratch . . . 1023
serif::dimensions::WithRespectTo< Quantity > . . . 1024
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace . . . 1024
serif::operators::backend::mfem::Workspace . . . 1026

Chapter 6

Class Index

6.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

serif::dimensions::Acceleration
: Dimension of Acceleration . . . 259

serif::discretization::blocks::rotation::angular_velocity
: The angular_velocity term represents the angular velocity of the system. It is a global scalar that is associated with the rotation of the system. This is used to enforce the rotation condition in the weak form when we are in the rigid rotation case. Note that downstream the way that this currently works is to calculate the expected angular velocity given the current mass, geometry, density, and angular momentum and then prescribe, given those, what the angular velocity should be. Therefore, this does not enforce that the angular velocity must be constant over the entire solver run, but rather that the angular velocity is consistent with the current state of the system . . . 261

serif::dimensions::AngularMomentum
: Dimension of Angular Momentum . . . 263

serif::equations::AngularMomentum
: Close the unknown angular speed with integral_star rho Omega s^2 dV - target_momentum = 0 . . . 265

serif::dimensions::AngularVelocity
: Dimension of Angular Velocity . . . 269

serif::utils::dependencies::detail::AppendBinding< BindingsT, BindingT > . . . 270

serif::utils::dependencies::detail::AppendBinding< Bindings< BindingsTs... >, BindingT > . . . 270

serif::utils::dependencies::detail::AppendKeys< ResultT, Keys > . . . 271

serif::utils::dependencies::detail::AppendKeys< ResultT, FirstT, Rest... > . . . 271

serif::utils::dependencies::detail::AppendUnique< ListT, T > . . . 272

serif::utils::dependencies::detail::AppendUnique< KeyList< Keys... >, T > . . . 272

serif::dimensions::Area
: Dimension of Area . . . 273

serif::geometry::surface::detail::AreSurfaceBindings< Bindings > . . . 275

serif::geometry::surface::detail::AreSurfaceBindings< utils::dependencies::Bindings< Entries... > > . . . 276

serif::operators::detail::AsKeyList< ListT > . . . 276

serif::operators::detail::AsKeyList< utils::types::TypeList< T... > > . . . 277

serif::operators::BadContributionCollection . . . 277

serif::operators::BadContributionRow . . . 279
serif::discretization::blocks::barotropic_constant
The barotropic_constant field is a field that represents the barotropic constant of the system. It is a global scalar that is associated with the mass normalization term. This is used to enforce the mass normalization condition in the weak form when we are in the barotropic case. This field will not generalize as useful to the non-barotropic case. Note that we use compile time reflection to get the name of the field and the terms . . . 280
serif::equations::BarotropicClosure< EOST > . . . 282
serif::equations::BarotropicHydrostaticEquilibrium . . . 285
serif::operators::BarotropicHydrostaticEquilibrium . . . 287
serif::operators::backend::mfem::volume_keys::Basis< Contribution > . . . 290
serif::operators::backend::mfem::BasisBindings< ColumnCount > . . . 290
serif::discretization::backend::mfem::BasisEvaluation
Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application . . . 292
serif::operators::backend::mfem::BasisEvaluation
Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application . . . 300
serif::discretization::backend::mfem::BasisWorkspace
Reusable scratch storage for basis contractions, reducing repeated allocations . . . 307
serif::operators::backend::mfem::BasisWorkspace
Reusable scratch storage for basis contractions, reducing repeated allocations . . . 308
serif::utils::dependencies::Binding< KeyT, TargetT > . . . 309
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< K, T > 309
serif::utils::dependencies::Bindings< BindingTs > . . . 310
serif::discretization::blocks::block
Base class for all block types in the discretization framework . . . 310
serif::discretization::blocks::central_density::border
The border term represents the border row used to enforce a central density constraint. This is a global scalar as it is a single value over the entire domain . . . 312
serif::operators::backend::mfem::nodal_keys::BorderLinearization . . . 314
serif::utils::dependencies::detail::BorrowedValue< T > . . . 314
serif::discretization::domain::Boundary . . . 315
serif::discretization::forms::boundary_flux< TrialT, TestT >
The boundary_flux form represents the boundary flux operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the boundary flux operation . . . 316
serif::discretization::domain::ids::BoundaryDescriptor
The BoundaryDescriptor struct is a runtime representation of a boundary, containing its name and integer attribute ID. This is used to project compile-time BoundaryID types into runtime values . . . 318
serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >
The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation . . . 319
serif::discretization::domain::schema::BoundaryID< BoundaryT, IdentifierValue >
The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation . . . 321
serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs >
A list of boundary IDs in the discretization framework . . . 322

serif::discretization::domain::schema::BoundaryIDList< BoundaryIDTs >
A list of boundary IDs in the discretization framework . . . 323
serif::discretization::domain::BoundaryIDResolver< BoundaryT, BoundaryIDListT > . . . 324
serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< Bounda
325
serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > >
326
serif::discretization::domain::BoundarySet< BoundaryTs > . . . 327
serif::geometry::surface::BoundaryValues< Term >
Values of one scalar term at an ordered collection of boundary points . . . 327
serif::geometry::surface::detail::BoundaryValueTracking . . . 329
serif::utils::dependencies::BoundType< BindingsT, KeyT > . . . 330
serif::utils::dependencies::detail::BoundType< BindingsT, KeyT > . . . 331
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT > . . . 331
serif::utils::dependencies::detail::CachedResult< ValueT, InputCount > . . . 332
serif::eos::relations::CatalogContainsRelation< Catalog, RelationT > . . . 333
serif::eos::relations::CatalogContainsRelation< RelationCatalog< Relations... >, RelationT > . . 334
serif::equations::CenteringConstraint . . . 335
serif::discretization::blocks::central_density
Field used to inject a border row used to enforce a central density constraint . . . 337
serif::equations::CentralDensityConstraint< EOST >
Prescribe central density through the continuous enthalpy: h(0) - h_EOS(rho_c) = 0 . . 339
serif::geometry::deformation::backend::mfem::volume_displacement_keys::CheckedBinding . . . 343
serif::dimensions::ChemicalPotential
Dimension of Chemical Potential . . . 343
serif::geometry::deformation::backend::mfem::volume_displacement_keys::Coefficients . . . 345
serif::discretization::blocks::surface_deformation::coefficients
The coefficients term represents some set of coefficients associated with whatever surface representation is being used to represent the surface deformation. Note how we do not have a functional space, support, or a quantity. This is also where we associate the values with 345
serif::discretization::backend::mfem::CoefficientTracking
Track coefficient values using an owned snapshot and the identity of their binding . . . 347
serif::operators::backend::mfem::CoefficientTracking
Track coefficient values using an owned snapshot and the identity of their binding . . . 350
serif::operators::backend::mfem::CoefficientView< BindingT > . . . 353
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::collector . . . 354
serif::operators::detail::Columns< ListT > . . . 355
serif::operators::detail::Columns< utils::types::TypeList< O... > > . . . 355
serif::operators::backend::mfem::CompactificationCoefficients . . . 356
serif::mapping::CompactificationPointData . . . 356
serif::utils::CompensatedSum
Neumaier summation retains small terms when a scalar reduction contains cancellation . 356
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >
Compose shape-to-surface and surface-to-volume maps with the complete derivative chain rule . . . 358
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT > . . . 363
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT > . . . 365
serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, ResourcesT, IndicesT >
372
serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::
373
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, Valid > . 374
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true > . 374
serif::utils::dependencies::ComputationGraph< Computations > . . . 376

serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, Independent > 378
serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, true > 379
serif::utils::dependencies::ComputationStore< EntryTs >
Own independent computation caches under compile-time role keys; this store does not connect their graphs 380
serif::utils::dependencies::ComputationUpdate< InputsT, ValueT > 383
serif::utils::misc::concepts::Concat< Lists > 384
serif::utils::misc::concepts::Concat< types::TypeList< LhsTs... >, types::TypeList< RhsTs... >, Rest... > 384
serif::utils::misc::concepts::Concat< types::TypeList< Ts... > > 385
serif::utils::misc::concepts::Concat<> 385
serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics . 386
serif::discretization::domain::schema::validation::RelationValidationResult::ConnectedDiagnostics . 386
serif::equations::Contribution< TestOperandT, SupportT >
Optional authoring base: a test operation and the domains of one weak contribution . . . 387
serif::operators::ContributionBinding< EquationT >
Retain one leaf equation and the constant coefficient of its arithmetic expression 388
serif::operators::detail::ContributionChildren< EquationT > 389
serif::operators::backend::mfem::ContributionData< OperandCount > 391
serif::operators::detail::ContributionOperands< TupleT > 391
serif::operators::detail::ContributionOperands< std::tuple< C... > > 392
serif::equations::Contributions< T, typename >
Collect leaf equation types without copying the objects in a composition 393
serif::equations::Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions())> > 393
serif::equations::ContributionSupport< T, typename > 394
serif::equations::ContributionSupport< T, std::void_t< typename T::Support > > 395
serif::equations::ContributionTuple< TupleT, typename > 395
serif::equations::ContributionTuple< std::tuple< Ts... >, std::void_t< decltype(std::tuple_cat(std::declval< typename C 396
serif::utils::dependencies::keys::Coordinates
Positions supplied to a computation, in the caller's coordinate convention 397
serif::discretization::domain::CoreDomain 397
serif::operators::detail::CouplingForm< Input, TestOperand > 398
serif::utils::types::ct_error_string< N > 399
serif::utils::types::ctstring< N > 400
serif::discretization::forms::Operation::Curl
Operation tag for evaluating a curl 401
serif::geometry::deformation::DeformationFailure
Describe a rejected geometry calculation; the dependency cache publishes results only after success 402
serif::geometry::deformation::DeformationLinearization< Map >
Freeze the shape coordinates associated with a derivative, independently of mutable caller inputs 403
serif::geometry::deformation::backend::mfem::DeformationValueTracking
Model options and coefficient copies describe values independently of wrapper addresses . 404
serif::geometry::deformation::DeformationWorkspaceType< Map, Linear >
Select numerical workspace storage without adding a preparation owner 406
serif::geometry::deformation::DeformationWorkspaceType< Map, true > 407
serif::utils::types::DenseMatrixBase< T > 407
serif::utils::types::DenseTensorBase< T, Rank > 408

serif::discretization::blocks::density
The density field used to represent the distribution of mass in the system. This is a scalar field that is associated with the mass density of the system. The density field is used to enforce the mass conservation condition in the weak form . . . 409
serif::equations::gravity_contributions::DensitySource . . . 411
serif::utils::dependencies::DependencyContext< ResourceTs > . . . 413
serif::utils::dependencies::DependencyContextView< ParentT, KeysT > . . . 415
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > > . . . 415
serif::dimensions::Dimensionless
Non dimensional type . . . 418
serif::geometry::surface::DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation >
Enforce P_EOS(state) - P_surface = 0 without inventing an inverse EOS relation . . . 419
serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >
Compile-time descriptor for a discrete functional space. The difference between a FunctionalSpace and a discrete functional space is that a discrete functional space carries some additional information which is needed to actually construct a finite element space (e.g. polynomial order, and rank). Once you have a discrete functional space you have, for MFEM at least, enough information to fully define a finite element space . . . 421
serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< FunctionalSpaceT >
422
serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< functional_space::DiscreteFunctionalSp
423
serif::discretization::blocks::displacement
The displacement field is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain . . . 424
serif::operators::backend::mfem::DisplacementCoefficients . . . 425
serif::equations::DisplacementForce< EOST >
Mechanical balance: pressure force + gravity force + rotational force . . . 426
serif::discretization::forms::Operation::Divergence
Operation tag for evaluating a divergence . . . 428
serif::discretization::forms::divergence< TrialT, TestT >
The divergence form represents the divergence operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the divergence operation . . . 428
serif::mapping::DMFixedMeshGeometry . . . 431
serif::mapping::DMGeometryCache . . . 431
serif::mapping::DMGeometryCacheStatistics . . . 433
serif::mapping::DMReferenceDisplacementSample . . . 433
serif::mapping::DMWorkspace . . . 434
serif::discretization::backend::mfem::DofMap
A mapping from a full MFEM finite-element space to a reduced set of DOFs that are supporter by the given element support. That is to say that if one has some set of dofs D (perhaps it looks like [0, 1, 2, 3, 4] as a hypothetical) and dofs 0, 2, and 4 are associated to domain A while 1 and 3 are associated to domain B. Then if one were to construct a DofMap with element support [1, 0, 1, 0, 1] (where 1 indicates that the element is supported and 0 indicates that it is not) then the reduced dof map would be [0, 2, 4] and the full to reduced map would be [-1, -1, 0, -1, 1] . . . 437
serif::discretization::backend::mfem::DofSelection
A selection of supported DOFs from a term binding, with boundary attribute information. The purpose here is to allow for the selection of a subset of DOFs from a term binding, such as those associated with a particular boundary, while still retaining the ability to validate the selection against the underlying mesh and term binding . . . 445
serif::discretization::domain::Domain . . . 449
serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs > . . . 450

serif::discretization::domain::schema::validation::DomainBoundary< BoundaryT, DomainTs > . . 452
serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics 454
serif::discretization::domain::schema::validation::RelationValidationResult::DomainBoundaryDiagnostics 455
serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >
State how the stellar surface moves and how that motion extends into the two surrounding regions . . . 457
serif::discretization::domain::ids::DomainDescriptor
The DomainDescriptor struct is a runtime representation of a domain, containing its name and integer attribute ID. This is used to project compile-time DomainID types into runtime values . . . 458
serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >
The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation 460
serif::discretization::domain::schema::DomainID< DomainT, IdentifierValue >
The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation 461
serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >
A list of domain IDs in the discretization framework . . . 462
serif::discretization::domain::schema::DomainIDList< DomainIDTs >
A list of domain IDs in the discretization framework . . . 463
serif::discretization::domain::DomainIDResolver< DomainT, DomainIDListT > . . . 465
serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... 465
serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > > 466
serif::geometry::deformation::keys::DomainMap . . . 467
serif::mapping::DomainMapper . . . 467
serif::utils::user_options::DomainMapperOptions . . . 474
serif::discretization::domain::DomainOrSet . . . 475
serif::discretization::domain::DomainOrSetList< DomainTs > . . . 476
serif::discretization::domain::relation::DomainRelation . . . 477
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations > . . 477
serif::discretization::domain::DomainSet< DomainTs > . . . 481
serif::dimensions::DynamicViscosity
Dimension of Dynamic Viscosity . . . 482
serif::operators::backend::mfem::ElementBasis< OperandCount >
Fixed-mesh basis samples; independent of the current coefficient values and displacement 484
serif::mapping::ElementCompactificationData . . . 485
serif::operators::backend::mfem::ElementData< OperandCount >
Weighted equation data retains the exact basis, geometry, and samples used to compute it 486
serif::mapping::ElementDisplacementData . . . 488
serif::operators::backend::mfem::ElementGeometry< OperandCount >
Physical mapping and mapped bases for one frozen displacement . . . 490
serif::mapping::ElementMappingData . . . 491
serif::geometry::deformation::EmptyDeformationWorkspace
Stateless linear actions need no hidden mutable scratch . . . 492
serif::dimensions::Energy
Dimension of Energy . . . 493
serif::dimensions::EnergyDensity
Dimension of Energy Density . . . 494

serif::discretization::blocks::enthalpy
The enthalpy field is a scalar field is a C0 continuous scalar field. Note that this field likely will not survive the transition to the more general non-barotropic case. In the barotropic case the gradient of enthalpy is parallel to the gradient of the pressure. This make enthalpy a particularly useful quantity as we can represent the 0 pressure surface condition as the 0 enthalpy condition and we can break down mass constraint into a game of optimizing the Bernoulli constant . . . 496
serif::dimensions::Entropy
Dimension of Entropy . . . 498
serif::discretization::backend::mfem::TensorBasisCache::Entry
A struct that represents an entry in the TensorBasisCache. Each entry contains a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself . 499
serif::operators::backend::mfem::TensorBasisCache::Entry
A struct that represents an entry in the TensorBasisCache. Each entry contains a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself . 501
serif::utils::misc::concepts::EnumerateTypes< TypeList, Descriptor > . . . 502
serif::utils::misc::concepts::EnumerateTypes< types::TypeList< TypeTs... >, Descriptor > . . . 502
serif::discretization::domain::EnvelopeDomain . . . 503
serif::eos::EOSEvaluationError . . . 504
serif::geometry::surface::keys::EOSResponse . . . 505
serif::eos::runtime::EOSView . . . 506
serif::equations::EquationDifference< LeftT, RightT >
Subtract every weak contribution of the right equation from the left equation . . . 510
serif::geometry::surface::keys::EquationOfState . . . 512
serif::operators::EquationPlan< EquationT, OperandsT > . . . 512
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > > . . . 512
serif::equations::EquationPoint< OperandsT > . . . 515
serif::equations::EquationRegistry< EquationTs > . . . 516
serif::operators::EquationRegistry< EquationTs > . . . 518
serif::equations::EquationScale< EquationT >
Multiply an equation's weak contributions by a constant independent of state and position 520
serif::equations::EquationSum< FirstT, RestTs >
Add equations sharing a residual row; each retains its own test operation and support . . 522
serif::mapping::compactification::ExteriorDomainMap . . . 524
serif::mapping::compactification::ExteriorMapDirection . . . 526
serif::mapping::compactification::ExteriorMapInput . . . 526
serif::mapping::compactification::ExteriorMapResult . . . 527
serif::mapping::compactification::ExteriorMapVariation . . . 528
serif::discretization::domain::mesh::FaceElements . . . 529
serif::mapping::FaceMappingContext . . . 529
serif::mapping::FaceMappingVariation . . . 531
serif::mapping::FaceQuadratureContext . . . 532
serif::discretization::blocks::field
The field struct is a tag type that is used to identify a field in the discretization. It is used to group together terms that belong to the same field. Each field should have a unique name and a list of terms that belong to it . . . 533
serif::discretization::blocks::FieldDescriptor
Runtime descriptor for a field, containing its name. Two FieldDescriptors are equal if they have identical names . . . 534
serif::utils::misc::concepts::Filter< List, Predicate > . . . 535
serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate > . . . 536
serif::utils::misc::concepts::Filter< types::TypeList<>, Predicate > . . . 536

serif::geometry::deformation::FixedInfinityRadialExtension
Reduce surface motion linearly to zero at reference infinity in logical reference radius . . 537
serif::dimensions::Force
Dimension of Force 538
serif::discretization::forms::force< TrialT, TestT >
The force form represents the force term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the force operation 539
serif::discretization::forms::form
The form struct is a tag type that is used to identify a form in the discretization. It is used to group together forms that belong to the same mathematical operation. Each form should have a unique name and a list of operand types that belong to it 542
std::formatter< serif::discretization::blocks::FieldDescriptor >
Specialization of std::formatter for FieldDescriptor and TermDescriptor to enable formatting with std::format 543
std::formatter< serif::discretization::blocks::TermDescriptor >
Specialization of std::formatter for TermDescriptor to enable formatting with std::format 544
std::formatter< serif::discretization::domain::DomainOrSet > 546
std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult > 547
std::formatter< serif::discretization::forms::FormDescriptor >
Overload to allow println for FormDescriptors 548
std::formatter< serif::discretization::functional_space::Continuity >
Specialization of the std::formatter for the Continuity enum. This allows us to use std↩ ::format to format the Continuity enum as strings 550
std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > >
Specialization of the std::formatter for the DiscreteFunctionalSpace types. This allows us to use std::format to format the DiscreteFunctionalSpace types as strings 551
std::formatter< serif::discretization::functional_space::Family >
Specialization of the std::formatter for the Family enum. This allows us to use std::format to format the Family enum as strings 553
std::formatter< serif::discretization::functional_space::Mapping >
Specialization of the std::formatter for the Mapping enum. This allows us to use std::format to format the Mapping enum as strings 554
std::formatter< serif::discretization::functional_space::Rank< R > >
Specialization of the std::formatter for the functional space types. This allows us to use std::format to format the functional space types as strings 556
std::formatter< serif::discretization::functional_space::SpaceDescriptor >
Specialization of the std::formatter for the SpaceDescriptor types. This allows us to use std::format to format the SpaceDescriptor types as strings 557
std::formatter< serif::discretization::functional_space::Trace >
Specialization of the std::formatter for the Trace enum. This allows us to use std::format to format the Trace enum as strings 559
std::formatter< serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext > . . 560
std::formatter< serif::discretization::quadrature::FormRule > 561
std::formatter< serif::discretization::quadrature::MappingKind > 563
std::formatter< serif::discretization::quadrature::Mode > 564
std::formatter< serif::discretization::quadrature::QuadratureRole > 565
std::formatter< serif::discretization::quadrature::Query > 567
std::formatter< serif::discretization::quadrature::RuleControl > 568
std::formatter< serif::discretization::quadrature::RuleSet > 569
serif::discretization::forms::FormDescriptor
Runtime form descriptor including runtime term information. Used to represented the compile-time form type at runtime 571
serif::discretization::quadrature::FormRule 573

serif::discretization::forms::FormTraits< FormT >
Compile-time traits for forms, used to extract types and properties from template parameters. This struct is specialized for each form type to provide specific traits such as trial and test order reductions . . . 574
serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > >
Specialization of FormTraits for the boundary_flux form, which has no order reduction for either trial or test terms . . . 575
serif::discretization::forms::FormTraits< divergence< TrialT, TestT > >
Specialization of FormTraits for the divergence form, which has an order reduction of 1, to satisfy the Inf-Sup condition, for the trial term and no order reduction for the test term . . . 576
serif::discretization::forms::FormTraits< force< TrialT, TestT > >
Specialization of FormTraits for the force form, which has no order reduction for either trial or test terms . . . 577
serif::discretization::forms::FormTraits< mass< TrialT, TestT > >
Specialization of FormTraits for the mass form, which has no order reduction for either trial or test terms . . . 578
serif::discretization::forms::FormTraits< projection< TrialT, TestT > >
Specialization of FormTraits for the projection form, which has no order reduction for either trial or test terms . . . 579
serif::discretization::forms::FormTraits< source< TrialT, TestT > >
Specialization of FormTraits for the source form, which has no order reduction for either trial or test terms . . . 581
serif::dimensions::Frequency
Dimension of Frequency . . . 582
serif::discretization::domain::relation::FullyConnected< A > . . . 584
serif::discretization::domain::schema::validation::FullyConnected< A > . . . 586
serif::discretization::blocks::generated_residual_block< GeneratedResidual >
Final class for generated residual blocks, derived from residual_block_base. These are blocks that are generated at runtime . . . 587
serif::discretization::blocks::generated_value_block< GeneratedValue >
Final class for generated value blocks, derived from value_block_base. These are blocks that are generated at runtime . . . 589
serif::operators::backend::mfem::volume_keys::Geometry< Contribution > . . . 591
serif::geometry::deformation::backend::mfem::GeometrySampleRule
Borrow one sampling rule; repeat an element to retain the actual union of downstream rules 591
serif::geometry::deformation::GeometryStepLimit
Geometry-only bound at the supplied sample points, with a location for the first obstruction 592
serif::geometry::deformation::GeometryStepOptions
Numerical controls for keeping a straight volume-displacement step above a Jacobian determinant floor . . . 594
serif::discretization::blocks::global_scalar
A global scalar is a term that has a single value over the entire domain (think angular momentum or total mass). Broadly we need to separate these from the other terms since they require different (often less enriched) information at compile time . . . 595
serif::discretization::forms::Operation::Gradient
Operation tag for evaluating a gradient . . . 596
serif::discretization::blocks::gravity::gradient
The gravitational gradient term is a 2nd order Hdiv vector field that is used to represent the negative gravitational acceleration of the system. Note that when I started developing this code I made the frankly wacky choice to let this be the true gradient of the potential, which is the negative acceleration. Broadly this may make sense mathematically; however, it means that developers must be extra careful to remember the sign convention . . . 596
serif::equations::gravity_contributions::GradientDivergence . . . 599
serif::operators::GradientRequest . . . 601

serif::equations::gravity_contributions::GradientValue . . . 603
serif::utils::dependencies::detail::Graph< Computations > . . . 605
serif::dimensions::GravitationalPotential
Dimension of Gravitational Potential . . . 607
serif::discretization::blocks::gravity
The gravity field is represented as a mixed system, in many ways this field is the motivation for the entire design of this file. Since it is a mixed system we have two separate terms that make it up, the potential and the gradient. These are not derived from one and other, rather they are weakly enforced to agree. This buys us an additional order of convergence in the potential which in tern allows us to use a lower resolution mesh . . . 609
serif::equations::GravityForce . . . 611
serif::equations::GravityGradientRelation . . . 614
serif::mapping::GridFunctionMappingEvaluator . . . 615
serif::utils::dependencies::detail::HasBinding< BindingsT, KeyT > . . . 620
serif::utils::dependencies::HasBinding< BindingsT, KeyT > . . . 620
serif::utils::dependencies::detail::HasBinding< Bindings< BindingTs... >, KeyT > . . . 621
serif::eos::runtime::HasRuntimeQuantityIdentifiers< RelationT > . . . 621
serif::eos::runtime::HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > > 622
serif::operators::HasScalarVolumeOperands< EquationT, OperandsT > . . . 623
serif::operators::HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > > 624
serif::eos::models::ImplementsPartialDerivative_t< EOS, RelationT, InputQuantity > . . . 625
serif::eos::models::ImplementsRelation_t< EOS, RelationT > . . . 626
serif::eos::models::ImplementsRelationCatalog_t< EOS, Output, Inputs > . . . 627
serif::eos::models::ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > > 628
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >
A calculation may reuse unchanged pieces of its previous immutable result . . . 629
serif::discretization::domain::InfinitySurfaceBoundary . . . 631
serif::utils::dependencies::keys::Input
The main input to a computation . . . 633
serif::discretization::domain::relation::Inscribed< A, B > . . . 633
serif::discretization::domain::schema::validation::Inscribed< A, B > . . . 635
serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics . 637
serif::discretization::domain::schema::validation::RelationValidationResult::InscribedDiagnostics . . 638
serif::geometry::deformation::keys::InteriorModel . . . 638
serif::dimensions::InternalEnergy
Dimension of Internal Energy . . . 639
serif::utils::dependencies::detail::Invocation< ComputationT, ArgumentsT, typename > . . . 640
serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< decltype(st 641
serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, typename > . . . 642
serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std::remove_cvref_t< EOSCand 643
serif::discretization::blocks::IsGlobalScalarTerm< T >
Helper struct to determine if a type is a global scalar term (GlobalScalarTerm) . . . 644
serif::geometry::surface::IsobaricSurface . . . 645
serif::discretization::blocks::IsPhysicalSpatialTerm< T >
Helper struct to determine if a type is a physical spatial term (PhysicalSpatialTerm) . . . 648
serif::dimensions::IsQuantityValue< Candidate > . . . 650
serif::dimensions::IsQuantityValue< QuantityValue< Quantity > > . . . 651
serif::discretization::blocks::IsSpatialTerm< T >
Helper struct to determine if a type is a spatial term (SpatialTerm) . . . 652
serif::eos::relations::IsThermodynamicRelation< RelationCandidate > . . . 653

serif::eos::relations::IsThermodynamicRelation< Relation< Output, Inputs... > > . . . 654
serif::eos::relations::IsValidRelationCatalog< CandidateCatalog > . . . 655
serif::eos::relations::IsValidRelationCatalog< RelationCatalog< Relations... > > . . . 656
serif::operators::backend::mfem::JacobianOperator< CacheT > . . . 657
serif::operators::detail::Join< Lists > . . . 660
serif::mapping::compactification::KelvinCompactification . . . 661
serif::utils::user_options::KelvinCompactificationOptions . . . 665
serif::utils::dependencies::KeyList< KeyTs >
An ordered collection of role keys, independent of their associated value types . . . 665
serif::utils::dependencies::KeyListBindings< BindingsT, KeysT > . . . 666
serif::utils::dependencies::KeyListBindings< BindingsT, KeyList< KeyTs... > > . . . 666
serif::utils::dependencies::detail::KeyListSize< KeysT > . . . 667
serif::utils::dependencies::detail::KeyListSize< KeyList< Keys... > > . . . 667
serif::dimensions::KinematicViscosity
Dimension of Kinematic Viscosity . . . 668
serif::dimensions::Length
Dimension of Length . . . 670
serif::geometry::deformation::keys::Linearization . . . 671
serif::geometry::surface::keys::Linearization . . . 672
serif::operators::backend::mfem::nodal_keys::Linearization . . . 672
serif::operators::backend::mfem::volume_keys::Linearization . . . 672
serif::geometry::deformation::backend::mfem::detail::LocatedBoundaryPoint
Keep the best inverse boundary-map result and its physical residual . . . 672
serif::dimensions::Luminosity
Dimension of Luminosity . . . 673
serif::dimensions::MagneticFluxDensity
Dimension of Magnetic Flux Density . . . 675
serif::operators::backend::mfem::MappingCoefficients . . . 676
serif::mapping::MappingPointContext . . . 676
serif::mapping::MappingPointVariation . . . 678
serif::dimensions::Mass
Dimension of mass . . . 679
serif::discretization::blocks::density::mass
The mass density term for the system. Note that the naming here could be a bit confusing as FEM also uses the term ”mass” to represent the canonical mass matrix. This is a more standard physics reading of what is going on here. We define that this space is a second order scalar L2 space. This allows for the possibility of discontinuous density fields . . . 681
serif::discretization::forms::mass< TrialT, TestT >
The mass form represents the mass matrix operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the mass operation . . . 684
serif::discretization::blocks::barotropic_constant::mass_normalization
Mass_normalization term represents the mass normalization condition of the system. That is that it enforces that the total mass of the system is equal to the target mass in a weak sense. This is both a term and a global scalar . . . 688
serif::dimensions::MassDensity
Dimension of Mass Density . . . 690
serif::dimensions::MassFlowRate
Dimension of Mass Flow Rate . . . 691
serif::equations::MassNormalization
Close the unknown Bernoulli constant with integral_star rho dV - target_mass = 0 . . . 693
serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, Catalog > . . . 695
serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Re
695

serif::utils::dependencies::detail::MergeKeyLists< ResultT, Lists > . . . 696
serif::utils::dependencies::detail::MergeKeyLists< ResultT, KeyList< Keys... >, Rest... > . . . 697
serif::operators::backend::mfem::MeshAndCoefficientBindings . . . 698
serif::discretization::domain::mesh::MeshTopology . . . 698
serif::operators::backend::mfem::MetadataTracking
Metadata is a value; moving its temporary wrapper does not change the mesh or bindings it describes . . . 702
serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< OperandListT > . . . 703
serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< utils::types::TypeList< OperandTs... > > 703
serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext . . . 704
serif::discretization::quadrature::backend::mfem::MFEMResolver . . . 706
serif::discretization::quadrature::backend::mfem::MFEMRule . . . 708
serif::dimensions::MomentOfInertia
Dimension of Moment Of Inertia . . . 709
serif::dimensions::Momentum
Dimension of Momentum . . . 710
serif::operators::NestedBadCollection . . . 712
serif::operators::backend::mfem::NodalData< EquationT, SchemaT >
A frozen nodal residual and derivative retain the selection needed to interpret every entry 713
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >
Fixed coefficient ordering and selected rows shared by the graph and its accepted results 714
serif::geometry::deformation::NodalRadialSurface
Give each surface node an independent radial displacement, measured in physical length 716
serif::operators::NodalRequest< TermT, OperationT, PositionIndependent > . . . 718
serif::operators::backend::mfem::NodalValueTracking
Moving the context or copying equal coefficients does not invalidate the nodal calculation 720
serif::discretization::forms::Operation::NormalTrace
Operation tag for evaluating the normal flux along a boundary . . . 721
serif::dimensions::NumberDensity
Dimension of Number Density . . . 722
serif::operators::backend::mfem::NumericalWork
Numerical work derived from the dependency engine's most recent report . . . 723
serif::dimensions::Opacity
Dimension of Opacity . . . 725
serif::discretization::forms::Operand< TermT, OperationT >
Compile time representation of some operand. An operand associates a term to an operation type (e.g. the gradient of gravitational potential where gravitational potential is the term and gradient is the operation) . . . 727
serif::operators::backend::mfem::OperandEvaluator< OperandT > . . . 728
serif::operators::backend::mfem::OperandEvaluator< OperandT >
Scalar values and H1/H(div) vector values or divergences share coefficient contractions . 728
serif::discretization::forms::OperandResult< OperandT >
General case for compile time metadata associating some operand to the value type of its result. These use the OperationValue struct and the information in the operand to determine the value type of the result of that operand . . . 733
serif::discretization::forms::OperandResult< OperandT >
Specialization of the OperandResult struct for operands that are valid operands (satisfy the IsOperand concept) and are spatial terms (e.g defined over the domain, think mass density). This specialization uses the OperationValue struct to determine the value type of the result of the operand based on its term type and operation type . . . 734
serif::operators::backend::mfem::OperandSamples
Reference samples remain available for derivatives through the physical mapping . . . 735
serif::equations::OperandValues< OperandsT, OutputT > . . . 736

serif::operators::OperandValues< OperandsT, OutputT > . . . 736
serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT > . . . 737
serif::discretization::forms::Operation
The Operation struct defines a set of operations that can be applied to terms in the context of a form. These operations include value, gradient, divergence, curl, and normal trace. Similar to the statement I made at the top of forms.hpp, if you are a developer looking to extend the mathematical capabilities of SERiF this is a place you should be looking early on 739
serif::discretization::forms::OperationValue< Rank, OperationT >
General case for compile time metadata associating a tensor rank to an operation. The specialization of this will also specific what the return value of an operation of that rank will be . . . 740
serif::discretization::forms::OperationValue< 0, Operation::Gradient >
The gradient of a rank 0 tensor is a rank 1 tensor (a vector of doubles) . . . 740
serif::discretization::forms::OperationValue< 0, Operation::Value >
The value of a rank 0 tensor is a scalar, a double in the case of SERiF where we restrict ourselves to double precision . . . 741
serif::discretization::forms::OperationValue< 1, Operation::Curl >
The curl of a rank 1 tensor is a rank 1 tensor (a vector of doubles) . . . 741
serif::discretization::forms::OperationValue< 1, Operation::Divergence >
The divergence of a rank one tensor is a scalar (a double) . . . 742
serif::discretization::forms::OperationValue< 1, Operation::NormalTrace > . . . 743
serif::discretization::forms::OperationValue< 1, Operation::Value >
The value of a rank 1 tensor is a rank 1 tensor (a vector of doubles in SERiF) . . . 743
serif::utils::dependencies::keys::Output
The main result of a computation . . . 744
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT > . . . 744
serif::utils::dependencies::detail::OwnedValue< T > . . . 748
serif::geometry::surface::IsobaricSurface::Parameters . . . 749
serif::utils::dependencies::keys::Parameters
Configuration values that can affect the result . . . 749
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >
Class to represent a compile time derivative relation between two quantities. Note that this class does not, in fact cannot, compute any derivatives itself. Rather, this is a way for the compiler to understand higher level mathematical relations. Unless you are really in the weeds of the development team you will likely not need to interact with this directly. Further note that this class does include (unlike the more basic dimension system) some basic dimensional analysis. Specifically, there are some overloaded operators that allow for certain specializations of PartialDerivative to operate on one and other conditioned on what their input and output types are. This does not do any thing like convert area to length^2. However, it will do more basic things like ensure that you can only add two partials which agree on their dimensions exactly. Or that while you can multiple a partial by any scalar, you cannot multiply arbitrary partials . . . 749
serif::mapping::PhysicalPositionFunctionCoefficient . . . 756
serif::dimensions::PhysicalQuantity . . . 758
serif::operators::backend::mfem::volume_keys::PhysicalValues< Contribution, OperandT > . . . 759
serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >
Store the scalar and point linearization together so a result cannot mix accepted states . 759
serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >
A point condition and its scalar border must use the same mesh . . . 760
serif::equations::PointLinearization< OperandsT, ResultT > . . . 762
serif::equations::PoissonBalance . . . 763
serif::discretization::quadrature::Policy . . . 765
serif::eos::models::Polytrope . . . 766
serif::eos::models::PolytropeParameters . . . 770

serif::discretization::blocks::gravity::potential
The gravitational potential term is a 2nd order L2 scalar field that is used to represent the gravitational potential of the system. Note that this is not exactly the same as the if you were to take the scalar potential of the gravitational gradient term. These are solved for by the code separately and weakly enforced to agree. Generally if you need the most accurate gradient you should use the gradient term and if you need the most accurate potential you should use this term (this is because differentiating this term loses an order of accuracy in the differentiation) . . . 770
serif::equations::gravity_contributions::PotentialValue . . . 774
serif::dimensions::Power
Dimension of Power . . . 776
serif::geometry::deformation::PowerLawRadialExtension
Extend surface motion toward a fixed stellar center using a power of logical reference radius 777
serif::utils::dependencies::detail::UsedResource< GraphT >::Predicate< ResourceT > . . . 778
serif::utils::dependencies::PreparationFailure< ErrorT > . . . 779
serif::operators::backend::mfem::PreparationRejection . . . 780
serif::utils::dependencies::PreparationReport< ResultsT, SourcesT > . . . 781
serif::utils::misc::concepts::Prepend< T, List > . . . 783
serif::utils::misc::concepts::Prepend< T, types::TypeList< Ts... > > . . . 783
serif::dimensions::Pressure
Dimension of Pressure . . . 784
serif::geometry::surface::keys::PressureCondition . . . 785
serif::geometry::surface::detail::PressureConditionTracking . . . 786
serif::equations::PressureForce< EOST >
Pressure's contribution to displacement balance: -integral_star P(h) div(w) dV . . . 787
serif::equations::PressureSurfaceConstraint< EOST >
Prescribed pressure on the stellar boundary: h - h_EOS(P_surface) = 0 . . . 790
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >
Common physical metadata; numerical values and preparation belong to the dependency context and cache . . . 793
serif::geometry::surface::PressureSurfaceDescriptor . . . 796
serif::geometry::surface::PressureSurfaceFormulation< EOS, Bindings, Carrier >
The boundary residual is carrier minus the EOS value at prescribed pressure; no carrier quantity is privileged . . . 797
serif::discretization::backend::mfem::TensorBasis::Product
A struct that represents a product of one-dimensional basis functions . . . 799
serif::utils::dependencies::ProjectedSnapshot< FunctionT > . . . 800
serif::discretization::forms::projection< TrialT, TestT >
The projection form represents the projection operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the projection operation . . . 801
serif::operators::backend::mfem::volume_keys::QuadratureLinearization< Contribution > . . . 804
serif::discretization::quadrature::QuadratureManifestOptions . . . 804
serif::discretization::quadrature::QuadratureOptions . . . 805
serif::discretization::quadrature::QuadratureRoleOptions . . . 809
serif::discretization::quadrature::QuadratureTermOptions . . . 810
serif::discretization::quadrature::QuadratureValidationOptions . . . 811
serif::dimensions::QuantityAt< Index, Quantities > . . . 812
serif::dimensions::QuantityAt< Index, QuantityList< Quantities... > > . . . 812
serif::dimensions::QuantityList< Quantities > . . . 812
serif::dimensions::QuantityOf< Candidate > . . . 812
serif::dimensions::QuantityOf< QuantityValue< Quantity > > . . . 813

serif::dimensions::QuantityValue< Quantity >
This is the non derivative equivalent to PartialDerivative. This associates a dimension with a value and, if that value can be determined at compile time, can be decomposed to a transparent double through .value() 813
serif::discretization::quadrature::Query 820
serif::geometry::deformation::RadialExtension
Extend one interleaved surface vector into the MFEM component-block volume layout. Stellar ownership wins at shared nodes. Both actions use the same scalar interpolation, so their pairing is Euclidean virtual work without an additional surface-area weight 822
serif::mapping::compactification::KelvinCompactification::RadialFactors 826
serif::geometry::deformation::keys::RadialInterpolation 826
serif::geometry::deformation::RadialInterpolation
Own the surface interpolation shared by the stellar and vacuum extensions 827
serif::discretization::functional_space::Rank< R >
Compile-time descriptor for the rank of a functional space 829
serif::geometry::deformation::backend::mfem::ReferenceDiscretization
Borrow the paired reference meshes; attributes retain their physical domain meaning . . 830
serif::geometry::deformation::keys::ReferenceDiscretization 832
serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot
One comparable snapshot covers all inputs used to compile fixed deformation data 832
serif::geometry::deformation::backend::mfem::ReferenceDiscretizationTracking
Let the generic dependency engine track MFEM metadata without owning meshes 834
serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot
Capture coordinate revisions as well as topology and mutable material/boundary labels . 835
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot
Track the coefficient interpretation independently of numerical coefficient values 837
serif::geometry::deformation::keys::ReferenceSurface 838
serif::geometry::deformation::ReferenceSurface
Own fixed surface positions and their canonical ascending scalar-node indices 839
serif::operators::backend::mfem::volume_keys::ReferenceValues< Contribution, OperandT > . . . 840
serif::eos::relations::Relation< Output, Inputs > 840
serif::eos::relations::RelationCatalog< Relations > 841
serif::eos::relations::RelationContainsInput< RelationT, Quantity > 842
serif::eos::relations::RelationContainsInput< Relation< Output, Inputs... >, Quantity > 843
serif::discretization::domain::relation::lists::RelationList< RelationTs > 844
serif::discretization::domain::schema::RelationList< RelationTs > 844
serif::geometry::surface::detail::RelationQuantities< Relation > 845
serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > > 845
serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, Relation 846
serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, lists::Re 846
serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< RelationT, DomainIDsT, BoundaryIDs 847
serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< DomainBoundary< BoundaryT, Doma 847
serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< FullyConnected< DomainT >, Domai 848
serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< Inscribed< InnerDomainT, OuterDom 849
serif::discretization::domain::relation::validation::RelationValidationResult 849
serif::discretization::domain::schema::validation::RelationValidationResult 851
serif::discretization::domain::schema::validation::RelationValidator< RelationT > 852

serif::discretization::domain::schema::validation::RelationValidator< DomainBoundary< BoundaryT, DomainTs... > > 852
serif::discretization::domain::schema::validation::RelationValidator< FullyConnected< DomainT > > 853
serif::discretization::domain::schema::validation::RelationValidator< Inscribed< InnerT, OuterT > > 854
serif::discretization::blocks::density::mass::residual
The residual block is a block that contains the residual of the mass density term. It is a scalar field that is associated with the mass density of the system . . . 854
serif::discretization::blocks::displacement::vector::residual
The residual block is a block that contains the residual of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces . . . 856
serif::discretization::blocks::enthalpy::specific::residual
The residual block is a block that contains the residual of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system . . . 857
serif::discretization::blocks::global_scalar::residual
The residual block is a block that contains the residual of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the residual of the mass of the system (i.e. the difference between the current mass and the target mass) . . . 859
serif::discretization::blocks::gravity::gradient::residual
The residual block is a block that contains the residual of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system . . . 861
serif::discretization::blocks::gravity::potential::residual
The residual block is a block that contains the residual of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system . . . 862
serif::discretization::blocks::surface_deformation::shape_equilibrium::residual
The residuals of the shape equilibrium condition . . . 864
serif::discretization::blocks::residual_block< index_value >
Final class for residual blocks with a fixed index, derived from residual_block_base . . . 865
serif::discretization::blocks::residual_block_base
Base class for residual blocks, derived from block . . . 868
serif::discretization::quadrature::Resolution . . . 869
serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, Available > . . . 870
serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, true > . . . 871
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, Done > . . . 872
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, false > . . . 872
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true > . . . 873
serif::utils::dependencies::ResolveKeys< BindingsT, KeysT > . . . 874
serif::utils::dependencies::ResolveKeys< BindingsT, KeyList< KeyTs... > > . . . 874
serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, Valid > 874
serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, true > 875
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT > . . . 876
serif::utils::dependencies::detail::ResourceKeys< ResourcesT > . . . 878
serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > > . . . 878
serif::utils::dependencies::detail::Result< T > . . . 879
serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > > . . . 880
serif::equations::RigidRotation . . . 882
serif::discretization::quadrature::RoleControls . . . 885

serif::discretization::blocks::rotation
The rotational field, represented as the angular velocity term, given that we currently only model rigid rotation this is a global scalar. A future extension which models differential rotation will likely need to modify this to some vector field or other more complex representation of the rotation . . . 886
serif::equations::RotationalForce
Rotational contribution to displacement balance: -integral_star rho grad(Psi) . w dV . . 888
serif::discretization::quadrature::RuleControl . . . 891
serif::discretization::quadrature::RuleSet . . . 892
serif::eos::runtime::RuntimeCatalogDispatch< EOS, CatalogT > . . . 894
serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > > 895
serif::eos::runtime::RuntimeCatalogIsSupported< CatalogT > . . . 896
serif::eos::runtime::RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > > . 897
serif::eos::runtime::RuntimeCatalogStorage< EOS, CatalogT > . . . 897
serif::eos::runtime::RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > > 898
serif::dimensions::runtime::RuntimeQuantityValue
Runtime resolved type for a quantity value. This holds both an ID and a value. This is the runtime equivalent to a QuantityValue. Note that where these differ is that QuantityValue is built with template types. This allows the compiler to resolve it to a plain numeric type and still do optimizations That is generally not the case with RuntimeQuantityValue . . 899
serif::dimensions::runtime::RuntimeRelationDescriptor . . . 900
serif::eos::runtime::RuntimeRelationQuantities< RelationT > . . . 901
serif::eos::runtime::RuntimeRelationQuantities< relations::Relation< OutputT, InputTs... > > . . 901
serif::eos::runtime::RuntimeRelationsAreSupported< QuantitiesAreIdentified, RelationTs > . . . 902
serif::eos::runtime::RuntimeRelationsAreSupported< true, RelationTs... > . . . 903
serif::eos::runtime::RuntimeRelationStorage< EOS, RelationT > . . . 904
serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > > 904
serif::utils::dependencies::keys::Scale
A multiplicative scale supplied as an input . . . 906
serif::discretization::domain::schema::validation::SchemaRelationValidationResult . . . 906
serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, RelationListT > 908
serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, relation::lists::RelationList< Relat 908
serif::discretization::domain::schema::validation::SchemaValidationResult . . . 908
serif::operators::backend::mfem::SelectedNodalValues
Selected coefficients are owned by the context; the selection has a stable shared identity . 910
serif::utils::errors::serif_error_printer< Message, Code > . . . 911
serif::discretization::blocks::surface_deformation::shape_equilibrium
The shape equilibrium term represents the shape equilibrium condition of the system. This is a term that is associated with the surface deformation field and is used to enforce the shape equilibrium condition in the weak form. This is where we associate the residuals with. Note how we do not have a functional space, support, or a quantity. This is because the shape equilibrium condition is a global condition that is not associated with any particular functional space, support, or quantity . . . 912
serif::geometry::deformation::keys::ShapeCoefficients . . . 914
serif::geometry::deformation::spaces::ShapeParameters
Independent coordinates belonging to the selected surface representation . . . 914
serif::discretization::forms::source< TrialT, TestT >
The source form represents the source term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the source operation . . . 915
serif::utils::dependencies::detail::SourceSnapshot< ResourceT > . . . 917

serif::discretization::functional_space::SpaceDescriptor
Compile-time descriptor for a functional space. This struct is used to describe the family, continuity, trace, and mapping of a functional space at compile time . . . 918
serif::utils::types::SparseMatrixBase< T > . . . 920
serif::utils::types::SparseTensorBase< T, Rank > . . . 920
serif::discretization::blocks::SpatialTermTraits< TermT >
Helper struct representing the traits of some spatial term type, including its functional space, support, family, order, and rank . . . 921
serif::discretization::blocks::enthalpy::specific
Specific enthalpy is the term we specifically work with rather than the total enthalpy of dof. In fact heuristically I would say that generally we should always work in specific quantities rather than total quantities as there is not a well defined amount of material at some dof 923
serif::dimensions::SpecificEnergy
Dimension of Specific Energy . . . 926
serif::dimensions::SpecificEnthalpy
Dimension of Specific Enthalpy . . . 928
serif::dimensions::SpecificEntropy
Dimension of Specific Entropy . . . 929
serif::dimensions::SpecificInternalEnergy
Dimension of Specific Internal Energy . . . 931
serif::geometry::deformation::SphericalHarmonicMode
Identify a real harmonic, with negative orders using sine and positive orders using cosine 932
serif::geometry::deformation::SphericalHarmonicSurface
Represent radial displacement lengths with normalized real spherical harmonics on the reference surface . . . 933
serif::discretization::domain::StellarSurfaceBoundary . . . 935
serif::discretization::blocks::surface_deformation
The surface corollary field to the full displacement field. This is in some ways more general than the others in so far as it is not pointwise defined over the domain. Rather, it is some general field of numbers, divorced from the underlying discrimination, that is used to represent the surface deformation of the system. This may be used, for example to represent spherical harmonic coefficients of the surface deformation. For more information see the surface and deformation systems . . . 936
serif::geometry::deformation::keys::SurfaceDisplacement . . . 938
serif::geometry::deformation::spaces::SurfaceDisplacement
Cartesian surface traces with components interleaved at each canonical node . . . 938
serif::geometry::deformation::SurfaceEmbedding
Own a linear conversion from radial displacement lengths to Cartesian surface displacements 939
serif::geometry::surface::SurfaceEOSResponse< Formulation >
An EOS value and all of its local derivatives, evaluated together at one state . . . 944
serif::geometry::surface::SurfaceFailure
A rejected thermodynamic candidate, with a sample index when evaluating a boundary batch . . . 945
serif::geometry::deformation::keys::SurfaceMap . . . 946
serif::geometry::deformation::keys::SurfaceMetric . . . 946
serif::geometry::deformation::keys::SurfaceModel . . . 947
serif::geometry::surface::detail::SurfaceOperands< Terms > . . . 947
serif::geometry::surface::detail::SurfaceOperands< utils::dependencies::KeyList< Terms... > > . . . 947
serif::dimensions::Temperature
Dimension of Temperature . . . 948
serif::discretization::backend::mfem::TensorBasis
A class that represents a tensor basis for a finite element. This class is used to evaluate the basis functions and their derivatives at a set of reference points . . . 949

serif::discretization::backend::mfem::TensorBasisCache
Share immutable tensor plans within one operator; copied rules are keyed by their contents 955
serif::operators::backend::mfem::TensorBasisCache
Share immutable tensor plans within one operator; copied rules are keyed by their contents 957
serif::discretization::backend::mfem::TensorWorkspace
A workspace for tensor contractions to avoid repeated allocations and pack memory (TODO: does this actual pack memory in a useful manner when using mfem vectors? Probably something to look into) 959
serif::discretization::blocks::term< FieldT >
The term struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Note that generally you should always use term rather than term_base as term associates to a particular field at compile time 961
serif::discretization::blocks::term_base
The term_base struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Generally you should always use term rather than term_base as term associates to a particular field at compile time 962
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >
A runtime binding between a term descriptor and its MFEM finite-element space. This is used to project down the compile time term descriptors into runtime reinterpretations which can then be used to extract things like the list of dofs, the element support, and the mesh. Note that this is a child of DofMap, therefor all term bindings are also dof maps . 964
serif::discretization::blocks::TermDescriptor
Runtime descriptor for a term, containing its associated field and name. Two Term↩ Descriptors are equal if they have identical fields and names 968
serif::equations::TestOperandTraits< T, typename > 970
serif::equations::TestOperandTraits< T, std::void_t< typename T::TestOperand > > 970
serif::dimensions::ThermodynamicQuantity 971
serif::dimensions::runtime::ThermodynamicQuantityID
Runtime resolvable ID for a thermodynamic quantity. Used in conjunction with concepts to resolve compile time thermodynamic types to runtime values 973
serif::dimensions::Time
Dimension of Time 975
serif::dimensions::Torque
Dimension of Torque 977
serif::utils::types::TypeList< Ts > 978
serif::geometry::surface::detail::UniqueTerms< Terms > 978
serif::geometry::surface::detail::UniqueTerms< utils::types::TypeList< Terms... > > 978
serif::utils::dependencies::detail::UsedResource< GraphT > 979
serif::discretization::domain::VacuumDomain 979
serif::geometry::deformation::keys::VacuumModel 980
serif::equations::ValidContributions< RowT, TupleT > 981
serif::equations::ValidContributions< RowT, std::tuple< C... > > 982
serif::discretization::forms::Operation::Value
Operation tag for evaluating a value 982
serif::discretization::blocks::density::mass::value
The value block is a block that contains the value of the mass density term. It is a scalar field that is associated with the mass density of the system 983
serif::discretization::blocks::displacement::vector::value
The value block is a block that contains the value of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces 984

serif::discretization::blocks::enthalpy::specific::value
The value block is a block that contains the value of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system 986
serif::discretization::blocks::global_scalar::value
The value block is a block that contains the value of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the actual value of the mass of the system 987
serif::discretization::blocks::gravity::gradient::value
The value block is a block that contains the value of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system 989
serif::discretization::blocks::gravity::potential::value
The value block is a block that contains the value of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system 991
serif::discretization::blocks::surface_deformation::coefficients::value
The actual values of the coefficients for whatever surface representation is being used . . 992
serif::discretization::blocks::value_block< index_value >
Final class for value blocks with a fixed index, derived from value_block_base 994
serif::discretization::blocks::value_block_base
Base class for value blocks, derived from block 996
serif::utils::dependencies::ValueSnapshot
Default tracking owns an exact value copy; noncopyable resources supply another policy . 998
serif::discretization::blocks::displacement::vector
Displacement vector term is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain. This is a vector field that is associated with the mapping of the domain from the reference to physical spaces. E.g. if there is some domain O_ref which n nodal positions x^i_ref then there is some other domain O_phys with n nodel positions such that x^i_phys = x^i_ref + d^i where d^i is the displacement vector at node i 1000
serif::utils::types::VectorBase< T > 1003
serif::dimensions::Velocity
Dimension of Velocity 1004
serif::dimensions::Volume
Dimension of Volume 1005
serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >
Stable equation ownership also retains temporary and move-only contribution trees . . . 1007
serif::operators::VolumeContributions< TupleT > 1008
serif::operators::VolumeContributions< std::tuple< C... > > 1009
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >
One accepted result contains ordered contribution batches and the assembled residual . . 1010
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >
Borrowed discretization objects must outlive their graph and its accepted results 1012
serif::geometry::deformation::keys::VolumeDisplacement 1014
serif::geometry::deformation::spaces::VolumeDisplacement
Displacement coefficients in the declared volume discretization 1014
serif::geometry::deformation::keys::VolumeExtension 1014
serif::mapping::VolumeMappingCache 1015
serif::mapping::VolumeMappingContext 1017
serif::mapping::VolumeMappingVariation 1018
serif::operators::VolumeOperands< C, ListT > 1019
serif::operators::VolumeOperands< C, utils::types::TypeList< O... > > 1020
serif::operators::backend::mfem::VolumeOptions
Numerical choices fixed when the graph is constructed 1021
serif::mapping::VolumeQuadratureContext 1022

serif::operators::backend::mfem::VolumeScratch
Scratch belongs to graph callables and is never a dependency source 1023
serif::dimensions::WithRespectTo< Quantity >
A type that is empty at runtime but allows for the developer to specify to the compiler what quantity the partial is with respect to 1024
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace . 1024
serif::operators::backend::mfem::Workspace
Reusable scratch for applying the accepted element linearization 1026

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

src/compile_time_checks/serif/dimensions/quantities.cpp . . . 1031
src/compile_time_checks/serif/discretization/blocks/fields.cpp . . . 1032
src/compile_time_checks/serif/discretization/forms/forms.cpp . . . 1032
src/compile_time_checks/serif/discretization/forms/operands.cpp . . . 1033
src/compile_time_checks/serif/discretization/quadrature/backend/mfem/mfem_resolver.cpp . . . 1033
src/compile_time_checks/serif/discretization/quadrature/backend/mfem/mfem_static_order.cpp 1034
src/compile_time_checks/serif/eos/evaluation.cpp . . . 1034
src/compile_time_checks/serif/eos/models/traits.cpp . . . 1035
src/compile_time_checks/serif/eos/runtime/details.cpp . . . 1036
src/compile_time_checks/serif/eos/runtime/views.cpp . . . 1037
src/compile_time_checks/serif/geometry/deformation.cpp . . . 1038
src/compile_time_checks/serif/geometry/surface.cpp . . . 1039
src/compile_time_checks/serif/operators/computation_owners.cpp . . . 1040
src/compile_time_checks/serif/operators/plan.cpp . . . 1041
src/compile_time_checks/serif/operators/serial_backend.cpp . . . 1042
src/compile_time_checks/serif/utils/dependencies/binding.cpp . . . 1043
src/compile_time_checks/serif/utils/dependencies/computation_graph.cpp . . . 1044
src/compile_time_checks/serif/utils/dependencies/computation_store.cpp . . . 1044
src/compile_time_checks/serif/utils/dependencies/dependency_context.cpp . . . 1045
src/compile_time_checks/serif/utils/dependencies/key_list.cpp . . . 1046
src/compile_time_checks/serif/utils/dependencies/mixed_errors.cpp . . . 1047
src/include/serif/dimensions/dimensions.hpp
File containing prue dimensions. Note that this is not intended to be (nor does it have the capability to be) a general dimensional analysis library. There is no understanding here that, for example, a dimension of length^2 is an area. Rather, the purpose of this is to force developers to be explicit about what values they are passing to what functions. Further, most of this system is built to be compile time constant. This means that the compiler can still apply many optimizations. Note that the dimensions here do not contain a value, rather these are just the concept of a dimension. To store a value you must use a Quantity and QuantityValue . . . 1048

src/include/serif/dimensions/partial.hpp
Compile time tools to represent a partial derivative of one quantity wrt. another quantity. Note that this system is not responsible for computing partials or in fact derivatives of any kind. Rather, it is responsible for forcing the developer — or compiler at least — to ask for a specific, explicit, partial and have to spell out what that is. That in turn gives the compiler enough information to do a lot of downstream things. A motivating example would be automatic verification of the provided EOS against all partials that might be needed to ensure they are all exported by the EOS 1051
src/include/serif/dimensions/quantities.hpp
This file contains the code to associate numeric data to a dimension. We call this a Quantity and a QuantityValue 1053
src/include/serif/dimensions/type_alias.hpp
This file contains type aliases for the dimensions and quantities used in the discretization. This is solely useful in so far as it means we do not need to type out the longer QuantityValue<Dimension> type for each of the dimensions and quantities used in the discretization. This is a convenience file that is used to make the code more readable and maintainable 1054
src/include/serif/dimensions/runtime/concepts.hpp
Runtime concept for a type that can be projected to a runtime type. This is useful when converting from the compile time type system to something that can be inspected at runtime 1281
src/include/serif/dimensions/runtime/runtime.hpp
Runtime usable types representing the compile time dimension type system. Note that generally this is not a dimensional analysis library. Though perhaps in the future it will grow to something like that. Rather, this system is intended to ensure that developers are explicit which what numeric values mean. There is more detail on this in the file comment for dimensions.hpp and quantities.hpp 1075
src/include/serif/discretization/backend/mfem/basis_evaluation.hpp
Basis evaluation, pointwise mapping, and transpose application using dense or tensor-product representations 1057
src/include/serif/discretization/backend/mfem/coefficient_tracking.hpp
Snapshot and comparison policy for MFEM coefficient views 1059
src/include/serif/discretization/backend/mfem/dof_map.hpp
This file contains the DofMap which allows for the full dof vector to be projected into a reduced space of dofs on supported elements for some field Generally, while much of the code here may be exercised, you will rarely see a DofMap be directly invoked. Rather, these are generally implicitly constructed through their child class term_binding 1060
src/include/serif/discretization/backend/mfem/dof_selection.hpp
This file contains the DofSelection which allows for the selection of a subset of DOFs from a term binding, with boundary attribute information 1062
src/include/serif/discretization/backend/mfem/mfem_backend.hpp
This is a utility file which allows one import to bring in all of the mfem discretization backend functionality. This file includes 1063
src/include/serif/discretization/backend/mfem/point_dofs.hpp
This file contains the implementation of function select_point_dofs. This function is used to select the components from a single H1 vertex in the fixed computational mesh. This is used to implement point constraints in the discretization (e.g. fixed center of mass) . . . 1064
src/include/serif/discretization/backend/mfem/tensor_basis.hpp
Implementation of fast tensor evaluation functions. These do not contribute physics but are used to accelerate evaluations 1066
src/include/serif/discretization/backend/mfem/term_binding.hpp
A runtime binding between some term descriptor and its MFEM finite-element space. This is used to project down the compile time term descriptors into runtime reinterpretations which can then be used to extract out things like the list of dofs, the element support, and the mesh 1068

src/include/serif/discretization/blocks/base.hpp
This file provides the base classes for residual and value blocks in the discretization framework . . . 1069
src/include/serif/discretization/blocks/blocks.hpp . . . 1070
src/include/serif/discretization/blocks/concepts.hpp
This file defines concepts for blocks in the discretization framework . . . 1283
src/include/serif/discretization/blocks/fields.hpp
Define the fields and their associated terms for the discretization. This is a key file to understand if you are to understand SERiF's numerics. Each of these field types should contain all the information needed for the compiler to generate the correct . . . 1071
src/include/serif/discretization/blocks/runtime.hpp
Runtime descriptors for fields and terms, used to project compile time types into runtime values . . . 1077
src/include/serif/discretization/blocks/traits.hpp
Compile-time traits for fields and terms, used to extract types and properties from template parameters . . . 1083
src/include/serif/discretization/domain/concepts.hpp . . . 1285
src/include/serif/discretization/domain/physical_domains.hpp . . . 1101
src/include/serif/discretization/domain/resolver.hpp . . . 1110
src/include/serif/discretization/domain/types.hpp . . . 1123
src/include/serif/discretization/domain/ids/boundary.hpp
This file defines the BoundaryID type, which is used to bind a semantic surface (a Boundary) to the integer boundary attribute MFEM stores in the mesh file . . . 1092
src/include/serif/discretization/domain/ids/concepts.hpp
This file defines concepts for domain and boundary IDs in the discretization framework. These are specifically used to validate that a type is a valid DomainID or BoundaryID . 1286
src/include/serif/discretization/domain/ids/domain.hpp
This file defines the DomainID type, which is used to bind a semantic volume (a Domain) to the integer element attribute MFEM stores in the mesh file . . . 1095
src/include/serif/discretization/domain/ids/lists/concepts.hpp
This file defines concepts for domain ID lists in the discretization framework . . . 1288
src/include/serif/discretization/domain/ids/lists/lists.hpp
This file defines the DomainIDList and BoundaryIDList types, specifically these types are used to project down compile time types to runtime values . . . 1097
src/include/serif/discretization/domain/ids/lists/traits.hpp
Compile-time traits for domain ID lists, used to extract types and properties from template parameters . . . 1085
src/include/serif/discretization/domain/ids/validation/concepts.hpp
This file defines concepts for domain ID lists in the discretization framework. Specifically we define here concepts used to validate traits of a domain . . . 1290
src/include/serif/discretization/domain/ids/validation/unique_ids.hpp
This file defines a utility function to check if a pack of values are all unique. This is used to validate that domain and boundary IDs are distinct . . . 1099
src/include/serif/discretization/domain/mesh/topology.hpp . . . 1100
src/include/serif/discretization/domain/relation/concepts.hpp . . . 1292
src/include/serif/discretization/domain/relation/relations.hpp . . . 1103
src/include/serif/discretization/domain/relation/lists/concepts.hpp . . . 1293
src/include/serif/discretization/domain/relation/lists/relation_list.hpp . . . 1102
src/include/serif/discretization/domain/relation/lists/traits.hpp . . . 1086
src/include/serif/discretization/domain/relation/validation/concepts.hpp . . . 1294
src/include/serif/discretization/domain/relation/validation/runtime.hpp . . . 1079
src/include/serif/discretization/domain/relation/validation/validation.hpp . . . 1108
src/include/serif/discretization/domain/schema/concepts.hpp . . . 1295
src/include/serif/discretization/domain/schema/domain_schema.hpp . . . 1111

src/include/serif/discretization/domain/schema/schemas.hpp . . . 1113
src/include/serif/discretization/domain/schema/traits.hpp . . . 1088
src/include/serif/discretization/domain/schema/utils.hpp . . . 1115
src/include/serif/discretization/domain/schema/validation/all.hpp . . . 1116
src/include/serif/discretization/domain/schema/validation/boundary.hpp . . . 1094
src/include/serif/discretization/domain/schema/validation/connected.hpp . . . 1117
src/include/serif/discretization/domain/schema/validation/inscribed.hpp . . . 1118
src/include/serif/discretization/domain/schema/validation/results.hpp . . . 1120
src/include/serif/discretization/domain/schema/validation/validator.hpp . . . 1121
src/include/serif/discretization/forms/concept.hpp
This file defines concepts for forms in the discretization framework . . . 1124
src/include/serif/discretization/forms/form_traits.hpp
Compile-time traits for forms, used to extract types and properties from template parameters . . . 1126
src/include/serif/discretization/forms/forms.hpp
This file defines the forms used in the discretization framework, including mass, divergence, source, force, boundary flux, and projection forms . . . 1128
src/include/serif/discretization/forms/operands.hpp
This file defines the Operand type, which represents a term and an operation to be applied to that term in the context of a form . . . 1131
src/include/serif/discretization/forms/runtime.hpp
File defining some utilities to project form information from compile time into runtime . 1081
src/include/serif/discretization/forms/value.hpp
Encode compile time information about the shape (rank) and data type of the result of various Operations . . . 1132
src/include/serif/discretization/functional_space/space.hpp
This file defines the functional space types used in the discretization framework, including compile-time descriptors and runtime descriptors . . . 1135
src/include/serif/discretization/quadrature/policy.hpp . . . 1145
src/include/serif/discretization/quadrature/query.hpp . . . 1146
src/include/serif/discretization/quadrature/rules.hpp . . . 1148
src/include/serif/discretization/quadrature/backend/order.hpp . . . 1143
src/include/serif/discretization/quadrature/backend/mfem/concepts.hpp . . . 1297
src/include/serif/discretization/quadrature/backend/mfem/configure.hpp . . . 1138
src/include/serif/discretization/quadrature/backend/mfem/mfem_resolver.hpp . . . 1139
src/include/serif/discretization/quadrature/backend/mfem/mfem_static_order.hpp . . . 1141
src/include/serif/eos/evaluation.hpp . . . 1212
src/include/serif/eos/exceptions.hpp . . . 1351
src/include/serif/eos/models/concepts.hpp . . . 1299
src/include/serif/eos/models/polytropic.hpp . . . 1150
src/include/serif/eos/models/traits.hpp . . . 1089
src/include/serif/eos/relations/catalog.hpp . . . 1151
src/include/serif/eos/relations/concepts.hpp . . . 1300
src/include/serif/eos/relations/relations.hpp . . . 1105
src/include/serif/eos/relations/traits.hpp . . . 1091
src/include/serif/eos/runtime/concepts.hpp . . . 1301
src/include/serif/eos/runtime/details.hpp . . . 1209
src/include/serif/eos/runtime/views.hpp . . . 1152
src/include/serif/equations/angular_momentum.hpp . . . 1154
src/include/serif/equations/arithmetic.hpp . . . 1155
src/include/serif/equations/barotropic_closure.hpp . . . 1156
src/include/serif/equations/centering.hpp . . . 1157
src/include/serif/equations/central_density.hpp . . . 1158
src/include/serif/equations/concepts.hpp . . . 1303

src/include/serif/equations/contributions.hpp . . . 1306
src/include/serif/equations/displacement_force.hpp . . . 1159
src/include/serif/equations/gravity_force.hpp . . . 1160
src/include/serif/equations/hydrostatic_equilibrium.hpp . . . 1161
src/include/serif/equations/mass_normalization.hpp . . . 1162
src/include/serif/equations/mixed_gravity.hpp . . . 1163
src/include/serif/equations/point.hpp . . . 1164
src/include/serif/equations/pressure_force.hpp . . . 1165
src/include/serif/equations/pressure_surface.hpp . . . 1166
src/include/serif/equations/registry.hpp . . . 1167
src/include/serif/equations/rigid_rotation.hpp . . . 1169
src/include/serif/equations/rotational_force.hpp . . . 1170
src/include/serif/equations/sum.hpp . . . 1171
src/include/serif/geometry/deformation/actions.hpp . . . 1171
src/include/serif/geometry/deformation/composition.hpp . . . 1188
src/include/serif/geometry/deformation/computations.hpp . . . 1175
src/include/serif/geometry/deformation/deformation.hpp . . . 1190
src/include/serif/geometry/deformation/failure.hpp . . . 1194
src/include/serif/geometry/deformation/geometry_step.hpp . . . 1181
src/include/serif/geometry/deformation/linearization.hpp . . . 1196
src/include/serif/geometry/deformation/models.hpp . . . 1199
src/include/serif/geometry/deformation/radial_extension.hpp . . . 1200
src/include/serif/geometry/deformation/reference_surface.hpp . . . 1201
src/include/serif/geometry/deformation/spaces.hpp . . . 1203
src/include/serif/geometry/deformation/surface_embedding.hpp . . . 1203
src/include/serif/geometry/deformation/backend/mfem/computations.hpp . . . 1174
src/include/serif/geometry/deformation/backend/mfem/geometry_step.hpp . . . 1179
src/include/serif/geometry/deformation/backend/mfem/inputs.hpp . . . 1182
src/include/serif/geometry/deformation/backend/mfem/radial_interpolation.hpp . . . 1183
src/include/serif/geometry/deformation/backend/mfem/reference_geometry.hpp . . . 1185
src/include/serif/geometry/deformation/backend/mfem/surface_metric.hpp . . . 1186
src/include/serif/geometry/deformation/backend/mfem/volume_displacement.hpp . . . 1187
src/include/serif/geometry/deformation/backend/mfem/detail/radial_locator.hpp . . . 1178
src/include/serif/geometry/deformation/detail/action_storage.hpp . . . 1190
src/include/serif/geometry/deformation/detail/determinant_polynomial.hpp . . . 1192
src/include/serif/geometry/deformation/detail/harmonics.hpp . . . 1193
src/include/serif/geometry/surface/compilation.hpp . . . 1205
src/include/serif/geometry/surface/computations.hpp . . . 1176
src/include/serif/geometry/surface/details.hpp . . . 1212
src/include/serif/geometry/surface/evaluation.hpp . . . 1215
src/include/serif/geometry/surface/isobaric.hpp . . . 1216
src/include/serif/geometry/surface/linearization.hpp . . . 1197
src/include/serif/geometry/surface/samples.hpp . . . 1219
src/include/serif/geometry/surface/surface.hpp
Isobaric conditions, EOS-selected formulations, and reusable boundary calculations . . . 1221
src/include/serif/geometry/surface/detail/batches.hpp . . . 1207
src/include/serif/geometry/surface/detail/evaluation.hpp . . . 1213
src/include/serif/geometry/surface/detail/relations.hpp . . . 1106
src/include/serif/mapping/coefficients.hpp . . . 1222
src/include/serif/mapping/context.hpp . . . 1230
src/include/serif/mapping/data.hpp . . . 1231
src/include/serif/mapping/domain_mapper.hpp . . . 1232
src/include/serif/mapping/evaluator.hpp . . . 1234
src/include/serif/mapping/status.hpp . . . 1235

src/include/serif/mapping/transformations.hpp . . . 1237
src/include/serif/mapping/variation.hpp . . . 1229
src/include/serif/mapping/volume_cache.hpp . . . 1239
src/include/serif/mapping/compactification/compactification.hpp . . . 1223
src/include/serif/mapping/compactification/input_result.hpp . . . 1224
src/include/serif/mapping/compactification/kelvin.hpp . . . 1226
src/include/serif/mapping/compactification/variation.hpp . . . 1228
src/include/serif/meta/enums.hpp . . . 1240
src/include/serif/meta/names.hpp . . . 1242
src/include/serif/operators/concepts.hpp . . . 1304
src/include/serif/operators/contributions.hpp . . . 1307
src/include/serif/operators/plan.hpp . . . 1309
src/include/serif/operators/backend/mfem/element_data.hpp . . . 1243
src/include/serif/operators/backend/mfem/element_evaluation.hpp . . . 1244
src/include/serif/operators/backend/mfem/element_preparation.hpp . . . 1246
src/include/serif/operators/backend/mfem/element_transpose.hpp . . . 1248
src/include/serif/operators/backend/mfem/jacobian_operator.hpp . . . 1250
src/include/serif/operators/backend/mfem/nodal_computations.hpp . . . 1251
src/include/serif/operators/backend/mfem/nodal_data.hpp . . . 1252
src/include/serif/operators/backend/mfem/nodal_evaluation.hpp . . . 1254
src/include/serif/operators/backend/mfem/nodal_inputs.hpp . . . 1255
src/include/serif/operators/backend/mfem/nodal_rows.hpp . . . 1257
src/include/serif/operators/backend/mfem/operand_evaluation.hpp . . . 1259
src/include/serif/operators/backend/mfem/point_border.hpp . . . 1261
src/include/serif/operators/backend/mfem/preparation.hpp . . . 1262
src/include/serif/operators/backend/mfem/quadrature.hpp . . . 1265
src/include/serif/operators/backend/mfem/rejection.hpp . . . 1266
src/include/serif/operators/backend/mfem/term_coefficients.hpp . . . 1268
src/include/serif/operators/backend/mfem/volume_application.hpp . . . 1269
src/include/serif/operators/backend/mfem/volume_computations.hpp . . . 1271
src/include/serif/operators/backend/mfem/volume_data.hpp . . . 1273
src/include/serif/operators/backend/mfem/volume_inputs.hpp . . . 1276
src/include/serif/operators/backend/mfem/volume_stages.hpp . . . 1277
src/include/serif/operators/backend/mfem/workspace.hpp . . . 1279
src/include/serif/utils/checks/finite.hpp . . . 1355
src/include/serif/utils/dependencies/binding.hpp . . . 1310
src/include/serif/utils/dependencies/computation.hpp . . . 1312
src/include/serif/utils/dependencies/computation_cache.hpp . . . 1314
src/include/serif/utils/dependencies/computation_graph.hpp . . . 1316
src/include/serif/utils/dependencies/computation_store.hpp . . . 1318
src/include/serif/utils/dependencies/dependencies.hpp
This file is a convenience header which includes all of the headers in the dependencies module. Further, we include the detailed documentation for the dependencies module here 1320
src/include/serif/utils/dependencies/dependency_context.hpp . . . 1324
src/include/serif/utils/dependencies/incremental_computation.hpp . . . 1336
src/include/serif/utils/dependencies/key_list.hpp . . . 1338
src/include/serif/utils/dependencies/key_list_operations.hpp . . . 1340
src/include/serif/utils/dependencies/keys.hpp . . . 1341
src/include/serif/utils/dependencies/preparation.hpp . . . 1264
src/include/serif/utils/dependencies/resource.hpp . . . 1342
src/include/serif/utils/dependencies/snapshot.hpp . . . 1344
src/include/serif/utils/dependencies/detail/cache_storage.hpp . . . 1325
src/include/serif/utils/dependencies/detail/cache_traits.hpp . . . 1327
src/include/serif/utils/dependencies/detail/graph.hpp . . . 1329

src/include/serif/utils/dependencies/detail/result.hpp . . . 1330
src/include/serif/utils/dependencies/detail/schema.hpp . . . 1332
src/include/serif/utils/dependencies/detail/type_lists.hpp . . . 1334
src/include/serif/utils/error/codes.hpp . . . 1345
src/include/serif/utils/error/errors.hpp . . . 1347
src/include/serif/utils/format/strings.hpp . . . 1363
src/include/serif/utils/misc/compensated_sum.hpp . . . 1348
src/include/serif/utils/misc/finite.hpp . . . 1356
src/include/serif/utils/misc/concepts/concat.hpp . . . 1349
src/include/serif/utils/misc/concepts/enumeration.hpp . . . 1350
src/include/serif/utils/misc/concepts/exceptions.hpp . . . 1352
src/include/serif/utils/misc/concepts/filter.hpp . . . 1353
src/include/serif/utils/misc/concepts/numeric.hpp . . . 1354
src/include/serif/utils/misc/concepts/type_uniqueness.hpp . . . 1361
src/include/serif/utils/misc/std_helper/cleaning.hpp . . . 1357
src/include/serif/utils/misc/std_helper/variadic_cleaning.hpp . . . 1358
src/include/serif/utils/misc/terminal/colors.hpp . . . 1359
src/include/serif/utils/misc/traits/type_uniqueness.hpp . . . 1362
src/include/serif/utils/types/strings.hpp . . . 1364
src/include/serif/utils/types/type_list.hpp . . . 1367
src/include/serif/utils/types/tensor/tensor.hpp . . . 1366
src/include/serif/utils/types/tensor/backend/mfem/tensor.hpp . . . 1365
src/include/serif/utils/user_options/mapping/domain_mapper.hpp . . . 1233
src/include/serif/utils/user_options/mapping/compactification/kelvin.hpp . . . 1227
src/include/serif/utils/warnings/warning.hpp . . . 1368
src/lib/serif/discretization/backend/mfem/basis_evaluation.cpp . . . 1370
src/lib/serif/discretization/domain/mesh/topology.cpp . . . 1370
src/lib/serif/discretization/domain/schema/validation/results.cpp . . . 1371
src/lib/serif/discretization/quadrature/policy.cpp . . . 1372
src/lib/serif/eos/models/polytropic.cpp . . . 1373
src/lib/serif/geometry/deformation/determinant_polynomial.cpp . . . 1379
src/lib/serif/geometry/deformation/harmonics.cpp . . . 1380
src/lib/serif/geometry/deformation/radial_extension.cpp . . . 1381
src/lib/serif/geometry/deformation/surface_embedding.cpp . . . 1381
src/lib/serif/geometry/deformation/backend/mfem/geometry_step.cpp . . . 1374
src/lib/serif/geometry/deformation/backend/mfem/inputs.cpp . . . 1375
src/lib/serif/geometry/deformation/backend/mfem/radial_interpolation.cpp . . . 1375
src/lib/serif/geometry/deformation/backend/mfem/radial_locator.cpp . . . 1376
src/lib/serif/geometry/deformation/backend/mfem/reference_geometry.cpp . . . 1377
src/lib/serif/geometry/deformation/backend/mfem/surface_metric.cpp . . . 1378
src/lib/serif/geometry/surface/isobaric.cpp . . . 1382
src/lib/serif/mapping/coefficients.cpp . . . 1383
src/lib/serif/mapping/data.cpp . . . 1384
src/lib/serif/mapping/domain_mapper.cpp . . . 1384
src/lib/serif/mapping/evaluator.cpp . . . 1385
src/lib/serif/mapping/transformations.cpp . . . 1385
src/lib/serif/mapping/volume_cache.cpp . . . 1387
src/lib/serif/mapping/compactification/kelvin.cpp . . . 1383

Chapter 8

Namespace Documentation

8.1 mfem Namespace Reference

8.2 serif Namespace Reference

Namespaces

- namespace dimensions
- namespace discretization
- namespace eos
- namespace equations
- namespace geometry
- namespace mapping
- namespace meta
- namespace operators
- namespace utils

8.3 serif::dimensions Namespace Reference

Namespaces

- namespace runtime

Classes

- struct Acceleration

  Dimension of Acceleration.
- struct AngularMomentum

  Dimension of Angular Momentum.
- struct AngularVelocity

  Dimension of Angular Velocity.
- struct Area

  Dimension of Area.
- struct ChemicalPotential

  Dimension of Chemical Potential.
- struct Dimensionless

  Non dimensional type.
- struct DynamicViscosity

  Dimension of Dynamic Viscosity.
- struct Energy

  Dimension of Energy.
- struct EnergyDensity

  Dimension of Energy Density.
- struct Entropy

  Dimension of Entropy.
- struct Force

  Dimension of Force.
- struct Frequency

  Dimension of Frequency.
- struct GravitationalPotential

  Dimension of Gravitational Potential.
- struct InternalEnergy

  Dimension of Internal Energy.
- struct IsQuantityValue
- struct IsQuantityValue< QuantityValue< Quantity > >
- struct KinematicViscosity

  Dimension of Kinematic Viscosity.
- struct Length

  Dimension of Length.
- struct Luminosity

  Dimension of Luminosity.
- struct MagneticFluxDensity

  Dimension of Magnetic Flux Density.
- struct Mass

  Dimension of mass.
- struct MassDensity

  Dimension of Mass Density.
- struct MassFlowRate

  Dimension of Mass Flow Rate.
- struct MomentOfInertia

Dimension of Moment Of Inertia.

- struct Momentum

  Dimension of Momentum.
- struct NumberDensity

  Dimension of Number Density.
- struct Opacity

  Dimension of Opacity.
- class PartialDerivative

  Class to represent a compile time derivative relation between two quantities. Note that this class does not, in fact cannot, compute any derivatives itself. Rather, this is a way for the compiler to understand higher level mathematical relations. Unless you are really in the weeds of the development team you will likely not need to interact with this directly. Further note that this class does include (unlike the more basic dimension system) some basic dimensional analysis. Specifically, there are some overloaded operators that allow for certain specializations of PartialDerivative to operate on one and other conditioned on what their input and output types are. This does not do any thing like convert area to length^2. However, it will do more basic things like ensure that you can only add two partials which agree on their dimensions exactly. Or that while you can multiple a partial by any scalar, you cannot multiply arbitrary partials.
- struct PhysicalQuantity
- struct Power

  Dimension of Power.
- struct Pressure

  Dimension of Pressure.
- struct QuantityAt
- struct QuantityAt< Index, QuantityList< Quantities... > >
- struct QuantityList
- struct QuantityOf
- struct QuantityOf< QuantityValue< Quantity > >
- class QuantityValue

  This is the non derivative equivalent to PartialDerivative. This associates a dimension with a value and, if that value can be determined at compile time, can be decomposed to a transparent double through .value()
- struct SpecificEnergy

  Dimension of Specific Energy.
- struct SpecificEnthalpy

  Dimension of Specific Enthalpy.
- struct SpecificEntropy

  Dimension of Specific Entropy.
- struct SpecificInternalEnergy

  Dimension of Specific Internal Energy.
- struct Temperature

  Dimension of Temperature.
- struct ThermodynamicQuantity
- struct Time

  Dimension of Time.
- struct Torque

  Dimension of Torque.
- struct Velocity

  Dimension of Velocity.
- struct Volume

  Dimension of Volume.
- struct WithRespectTo

  A type that is empty at runtime but allows for the developer to specify to the compiler what quantity the partial is with respect to.

Concepts

- concept PhysicalQuantityType

  Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity. Note all ThermodynamicQuantities will also satisfy this concept.
- concept ThermodynamicQuantityType

  Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity (both enforced through checking PhysicalQuantityType) and then that is is also derived from ThermodynamicQuantity. Note not all physical quanties will satisfy this.
- concept QuantityValueType

  Turn those compile time structs into a concept that can be used to constrain a template parameter.
- concept ThermodynamicQuantityValueType

Typedefs

- template<QuantityValueType Value>
  using QuantityOfT = typename QuantityOf<NeitherConstNorVolatileReference<Value>>::Type
- using DimensionlessValue = QuantityValue<Dimensionless>

  A dimensionless quantity, used for quantities that do not have a physical dimension (e.g., a pure number).
- using MassValue = QuantityValue<Mass>

  A quantity representing mass.
- using LengthValue = QuantityValue<Length>

  A quantity representing length.
- using TimeValue = QuantityValue<Time>

  A quantity representing time.
- using AreaValue = QuantityValue<Area>

  A quantity representing area.
- using VolumeValue = QuantityValue<Volume>

  A quantity representing volume.
- using DensityValue = QuantityValue<MassDensity>

  A quantity representing mass density.
- using NumberDensityValue = QuantityValue<NumberDensity>

  A quantity representing number density.
- using PressureValue = QuantityValue<Pressure>

  A quantity representing pressure.
- using TemperatureValue = QuantityValue<Temperature>

  A quantity representing temperature.
- using EntropyValue = QuantityValue<Entropy>

  A quantity representing entropy.
- using SpecificEntropyValue = QuantityValue<SpecificEntropy>

  A quantity representing specific entropy.
- using ChemicalPotentialValue = QuantityValue<ChemicalPotential>

  A quantity representing chemical potential.
- using EnergyValue = QuantityValue<Energy>

  A quantity representing energy..
- using InternalEnergyValue = QuantityValue<InternalEnergy>

  A quantity representing internal energy.
- using SpecificEnergyValue = QuantityValue<SpecificEnergy>

A quantity representing specific energy.

- using SpecificInternalEnergyValue = QuantityValue<SpecificInternalEnergy>

  A quantity representing specific internal energy.
- using SpecificEnthalpyValue = QuantityValue<SpecificEnthalpy>

  A quantity representing specific enthalpy.
- using EnergyDensityValue = QuantityValue<EnergyDensity>

  A quantity representing energy density.
- using GravitationalPotentialValue = QuantityValue<GravitationalPotential>

  A quantity representing gravitational potential.
- using VelocityValue = QuantityValue<Velocity>

  A quantity representing velocity.
- using AccelerationValue = QuantityValue<Acceleration>

  A quantity representing acceleration.
- using FrequencyValue = QuantityValue<Frequency>

  A quantity representing frequency.
- using AngularVelocityValue = QuantityValue<AngularVelocity>

  A quantity representing angular velocity.
- using MomentumValue = QuantityValue<Momentum>

  A quantity representing momentum.
- using AngularMomentumValue = QuantityValue<AngularMomentum>

  A quantity representing angular momentum.
- using MomentOfInertiaValue = QuantityValue<MomentOfInertia>

  A quantity representing moment of inertia.
- using ForceValue = QuantityValue<Force>

  A quantity representing force..
- using TorqueValue = QuantityValue<Torque>

  A quantity representing torque.
- using PowerValue = QuantityValue<Power>

  A quantity representing power.
- using LuminosityValue = QuantityValue<Luminosity>

  A quantity representing luminosity.
- using MassFlowRateValue = QuantityValue<MassFlowRate>

  A quantity representing mass flow rate.
- using OpacityValue = QuantityValue<Opacity>

  A quantity representing opacity.
- using DynamicViscosityValue = QuantityValue<DynamicViscosity>

  A quantity representing dynamic viscosity.
- using KinematicViscosityValue = QuantityValue<KinematicViscosity>

  A quantity representing kinematic viscosity.
- using MagneticFluxDensityValue = QuantityValue<MagneticFluxDensity>

  A quantity representing magnetic flux density.
- template<class T>
  using NeitherConstNorVolatile
- template<class T>
  using NeitherConstNorVolatileReference

8.3.1 Typedef Documentation

8.3.1.1 AccelerationValue

using serif::dimensions::AccelerationValue = QuantityValue<Acceleration>

A quantity representing acceleration.

8.3.1.2 AngularMomentumValue

using serif::dimensions::AngularMomentumValue = QuantityValue<AngularMomentum>

A quantity representing angular momentum.

8.3.1.3 AngularVelocityValue

using serif::dimensions::AngularVelocityValue = QuantityValue<AngularVelocity>

A quantity representing angular velocity.

8.3.1.4 AreaValue

using serif::dimensions::AreaValue = QuantityValue<Area>

A quantity representing area.

8.3.1.5 ChemicalPotentialValue

using serif::dimensions::ChemicalPotentialValue = QuantityValue<ChemicalPotential>

A quantity representing chemical potential.

8.3.1.6 DensityValue

using serif::dimensions::DensityValue = QuantityValue<MassDensity>

A quantity representing mass density.

8.3.1.7 DimensionlessValue

using serif::dimensions::DimensionlessValue = QuantityValue<Dimensionless>

A dimensionless quantity, used for quantities that do not have a physical dimension (e.g., a pure number).

8.3.1.8 DynamicViscosityValue

using serif::dimensions::DynamicViscosityValue = QuantityValue<DynamicViscosity>

A quantity representing dynamic viscosity.

8.3.1.9 EnergyDensityValue

using serif::dimensions::EnergyDensityValue = QuantityValue<EnergyDensity>

A quantity representing energy density.

8.3.1.10 EnergyValue

using serif::dimensions::EnergyValue = QuantityValue<Energy>

A quantity representing energy..

8.3.1.11 EntropyValue

using serif::dimensions::EntropyValue = QuantityValue<Entropy>

A quantity representing entropy.

8.3.1.12 ForceValue

using serif::dimensions::ForceValue = QuantityValue<Force>

A quantity representing force..

8.3.1.13 FrequencyValue

using serif::dimensions::FrequencyValue = QuantityValue<Frequency>

A quantity representing frequency.

8.3.1.14 GravitationalPotentialValue

using serif::dimensions::GravitationalPotentialValue = QuantityValue<GravitationalPotential>

A quantity representing gravitational potential.

8.3.1.15 InternalEnergyValue

using serif::dimensions::InternalEnergyValue = QuantityValue<InternalEnergy>

A quantity representing internal energy.

8.3.1.16 KinematicViscosityValue

using serif::dimensions::KinematicViscosityValue = QuantityValue<KinematicViscosity>

A quantity representing kinematic viscosity.

8.3.1.17 LengthValue

using serif::dimensions::LengthValue = QuantityValue<Length>

A quantity representing length.

8.3.1.18 LuminosityValue

using serif::dimensions::LuminosityValue = QuantityValue<Luminosity>

A quantity representing luminosity.

8.3.1.19 MagneticFluxDensityValue

using serif::dimensions::MagneticFluxDensityValue = QuantityValue<MagneticFluxDensity>

A quantity representing magnetic flux density.

8.3.1.20 MassFlowRateValue

using serif::dimensions::MassFlowRateValue = QuantityValue<MassFlowRate>

A quantity representing mass flow rate.

8.3.1.21 MassValue

using serif::dimensions::MassValue = QuantityValue<Mass>

A quantity representing mass.

8.3.1.22 MomentOfInertiaValue

using serif::dimensions::MomentOfInertiaValue = QuantityValue<MomentOfInertia>

A quantity representing moment of inertia.

8.3.1.23 MomentumValue

using serif::dimensions::MomentumValue = QuantityValue<Momentum>

A quantity representing momentum.

8.3.1.24 NeitherConstNorVolatile

template<class T>
using serif::utils::misc::std_helper::NeitherConstNorVolatile

8.3.1.25 NeitherConstNorVolatileReference

template<class T>
using serif::utils::misc::std_helper::NeitherConstNorVolatileReference

8.3.1.26 NumberDensityValue

using serif::dimensions::NumberDensityValue = QuantityValue<NumberDensity>

A quantity representing number density.

8.3.1.27 OpacityValue

using serif::dimensions::OpacityValue = QuantityValue<Opacity>

A quantity representing opacity.

8.3.1.28 PowerValue

using serif::dimensions::PowerValue = QuantityValue<Power>

A quantity representing power.

8.3.1.29 PressureValue

using serif::dimensions::PressureValue = QuantityValue<Pressure>

A quantity representing pressure.

8.3.1.30 QuantityOfT

template<QuantityValueType Value>
using serif::dimensions::QuantityOfT = typename QuantityOf<NeitherConstNorVolatileReference<Value>>::Type

8.3.1.31 SpecificEnergyValue

using serif::dimensions::SpecificEnergyValue = QuantityValue<SpecificEnergy>

A quantity representing specific energy.

8.3.1.32 SpecificEnthalpyValue

using serif::dimensions::SpecificEnthalpyValue = QuantityValue<SpecificEnthalpy>

A quantity representing specific enthalpy.

8.3.1.33 SpecificEntropyValue

using serif::dimensions::SpecificEntropyValue = QuantityValue<SpecificEntropy>

A quantity representing specific entropy.

8.3.1.34 SpecificInternalEnergyValue

using serif::dimensions::SpecificInternalEnergyValue = QuantityValue<SpecificInternalEnergy>

A quantity representing specific internal energy.

8.3.1.35 TemperatureValue

using serif::dimensions::TemperatureValue = QuantityValue<Temperature>

A quantity representing temperature.

8.3.1.36 TimeValue

using serif::dimensions::TimeValue = QuantityValue<Time>

A quantity representing time.

8.3.1.37 TorqueValue

using serif::dimensions::TorqueValue = QuantityValue<Torque>

A quantity representing torque.

8.3.1.38 VelocityValue

using serif::dimensions::VelocityValue = QuantityValue<Velocity>

A quantity representing velocity.

8.3.1.39 VolumeValue

using serif::dimensions::VolumeValue = QuantityValue<Volume>

A quantity representing volume.

8.4 serif::dimensions::runtime Namespace Reference

Classes

- struct RuntimeQuantityValue

  Runtime resolved type for a quantity value. This holds both an ID and a value. This is the runtime equivalent to a QuantityValue. Note that where these differ is that QuantityValue is built with template types. This allows the compiler to resolve it to a plain numeric type and still do optimizations That is generally not the case with RuntimeQuantityValue.
- struct RuntimeRelationDescriptor
- class ThermodynamicQuantityID

  Runtime resolvable ID for a thermodynamic quantity. Used in conjunction with concepts to resolve compile time thermodynamic types to runtime values.

Concepts

- concept RuntimeIdentifiedThermodynamicQuantity

Variables

- template<RuntimeIdentifiedThermodynamicQuantity Quantity>
  constexpr ThermodynamicQuantityID thermodynamicQuantityID {std::string_view{Quantity::identifier}}

  Extract a compile time identifier from a thermodynamic quantity and project it into a runtime string_view.

8.4.1 Variable Documentation

8.4.1.1 thermodynamicQuantityID

template<RuntimeIdentifiedThermodynamicQuantity Quantity>
ThermodynamicQuantityID serif::dimensions::runtime::thermodynamicQuantityID {std::string_view{Quantity::identifier}}
[inline], [constexpr]

Extract a compile time identifier from a thermodynamic quantity and project it into a runtime string_view.

Template Parameters

Quantity	Type which RuntimeIdentifiedThermodynamicQuantity allows

8.5 serif::discretization Namespace Reference

Namespaces

- namespace backend
- namespace blocks
- namespace domain
- namespace forms
- namespace functional_space
- namespace quadrature

8.6 serif::discretization::backend Namespace Reference

Namespaces

- namespace mfem

8.7 serif::discretization::backend::mfem Namespace Reference

Classes

- class BasisEvaluation

  Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application.
- struct BasisWorkspace

  Reusable scratch storage for basis contractions, reducing repeated allocations.
- struct CoefficientTracking

  Track coefficient values using an owned snapshot and the identity of their binding.
- class DofMap

  A mapping from a full MFEM finite-element space to a reduced set of DOFs that are supporter by the given element support. That is to say that if one has some set of dofs D (perhaps it looks like [0, 1, 2, 3, 4] as a hypothetical) and dofs 0, 2, and 4 are associated to domain A while 1 and 3 are associated to domain B. Then if one were to construct a DofMap with element support [1, 0, 1, 0, 1] (where 1 indicates that the element is supported and 0 indicates that it is not) then the reduced dof map would be [0, 2, 4] and the full to reduced map would be [-1, -1, 0, -1, 1].
- class DofSelection

  A selection of supported DOFs from a term binding, with boundary attribute information. The purpose here is to allow for the selection of a subset of DOFs from a term binding, such as those associated with a particular boundary, while still retaining the ability to validate the selection against the underlying mesh and term binding.
- class TensorBasis

  A class that represents a tensor basis for a finite element. This class is used to evaluate the basis functions and their derivatives at a set of reference points.
- class TensorBasisCache

  Share immutable tensor plans within one operator; copied rules are keyed by their contents.
- struct TensorWorkspace

  A workspace for tensor contractions to avoid repeated allocations and pack memory (TODO: does this actual pack memory in a useful manner when using mfem vectors? Probably something to look into).
- class TermBinding

  A runtime binding between a term descriptor and its MFEM finite-element space. This is used to project down the compile time term descriptors into runtime reinterpretations which can then be used to extract things like the list of dofs, the element support, and the mesh. Note that this is a child of DofMap, therefor all term bindings are also dof maps.

Enumerations

- enum class ReferenceOperation { value , gradient , value_and_gradient , divergence }

  Options enum to control what is actually evaluated at the reference points.
- enum class ContractionMode { automatic , dense }

  Options enum to control how the tensor contractions are performed.

Functions

- template<domain::IsBoundary BoundaryT, blocks::SpatialTerm TermT, domain::schema::IsSchema SchemaT>
  DofSelection select_boundary_dofs (const TermBinding< TermT, SchemaT > &binding)

  Select DOFs associated with a particular boundary from a term binding.
- template<blocks::SpatialTerm TermT, domain::schema::IsSchema SchemaT>
  requires (blocks::SpatialTermTraits<TermT>::family == functional_space::Family::h1)
  DofSelection select_point_dofs (const TermBinding< TermT, SchemaT > &binding, const std::array< double, 3 > &position={}, const double tolerance=1.0e-12)

  Select the dofs associated with a single point in the reference domain. This is used to implement point constraints in the discretization (e.g. fixed center of mass). Note that this function only works for H1 terms and will throw an error if used on a non-H1 term.

8.7.1 Enumeration Type Documentation

8.7.1.1 ContractionMode

enum class serif::discretization::backend::mfem::ContractionMode [strong]

Options enum to control how the tensor contractions are performed.

Enumerator

automatic	Automatically choose the contraction mode based on the number of points and dofs.
dense	Use a dense contraction mode.

8.7.1.2 ReferenceOperation

enum class serif::discretization::backend::mfem::ReferenceOperation [strong]

Options enum to control what is actually evaluated at the reference points.

Enumerator

value	Evaluate the basis functions at the reference points.
gradient	Evaluate the gradients of the basis functions at the reference points.
value_and_gradient	Evaluate both the basis functions and their gradients at the reference points.
divergence	Evaluate the divergence of the basis functions at the reference points.

8.7.2 Function Documentation

8.7.2.1 select_boundary_dofs()

template<domain::IsBoundary BoundaryT, blocks::SpatialTerm TermT, domain::schema::IsSchema SchemaT>
DofSelection serif::discretization::backend::mfem::select_boundary_dofs (
const TermBinding< TermT, SchemaT > & binding) [nodiscard]

Select DOFs associated with a particular boundary from a term binding.

Template Parameters

BoundaryT	The boundary type to select. This must be a valid boundary in the domain schema of the term binding.
TermT	The term type of the term binding. This must be a valid term type in the domain schema of the term binding. (e.g. something like Density)
SchemaT	The domain schema type of the term binding. This must be a valid domain schema type that contains the specified boundary.

Parameters

binding	The term binding from which to select DOFs. This must outlive the returned DOF selection.

Returns

A DofSelection object representing the selected DOFs associated with the specified boundary.

8.7.2.2 select_point_dofs()

```
template<blocks::SpatialTerm TermT, domain::schema::IsSchema SchemaT>
requires (blocks::SpatialTermTraits<TermT>::family == functional_space::Family::h1)
DofSelection serif::discretization::backend::mfem::select_point_dofs (
                const TermBinding< TermT, SchemaT > & binding,
                const std::array< double, 3 > & position = {},
                const double tolerance = 1.0e-12)   [nodiscard]
```

Select the dofs associated with a single point in the reference domain. This is used to implement point constraints in the discretization (e.g. fixed center of mass). Note that this function only works for H1 terms and will throw an error if used on a non-H1 term.

Template Parameters

TermT	The term type to select the dofs for. This must be a spatial term with an H1 functional space.
SchemaT	The schema type associated with the term.

Parameters

binding	The term binding to select the dofs from.
position	The position of the point in the reference domain.
tolerance	The tolerance for matching the point coordinates.

Returns

A DofSelection object representing the selected dofs.

8.8 serif::discretization::blocks Namespace Reference

Classes

- struct barotropic_constant

  The barotropic_constant field is a field that represents the barotropic constant of the system. It is a global scalar that is associated with the mass normalization term. This is used to enforce the mass normalization condition in the weak form when we are in the barotropic case. This field will not generalize as useful to the non-barotropic case. Note that we use compile time reflection to get the name of the field and the terms.
- struct block

  Base class for all block types in the discretization framework.
- struct central_density

  field used to inject a border row used to enforce a central density constraint
- struct density

  The density field used to represent the distribution of mass in the system. This is a scalar field that is associated with the mass density of the system. The density field is used to enforce the mass conservation condition in the weak form.
- struct displacement

  The displacement field is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain.
- struct enthalpy

  The enthalpy field is a scalar field is a C0 continuous scalar field. Note that this field likely will not survive the transition to the more general non-barotropic case. In the barotropic case the gradient of enthalpy is parallel to the gradient of the pressure. This make enthalpy a particularly useful quantity as we can represent the 0 pressure surface condition as the 0 enthalpy condition and we can break down mass constraint into a game of optimizing the Bernoulli constant.
- struct field

  The field struct is a tag type that is used to identify a field in the discretization. It is used to group together terms that belong to the same field. Each field should have a unique name and a list of terms that belong to it.
- struct FieldDescriptor

  Runtime descriptor for a field, containing its name. Two FieldDescriptors are equal if they have identical names.
- struct generated_residual_block

  Final class for generated residual blocks, derived from residual_block_base. These are blocks that are generated at runtime.
- struct generated_value_block

  Final class for generated value blocks, derived from value_block_base. These are blocks that are generated at runtime.
- struct global_scalar

  A global scalar is a term that has a single value over the entire domain (think angular momentum or total mass). Broadly we need to separate these from the other terms since they require different (often less enriched) information at compile time.
- struct gravity

  The gravity field is represented as a mixed system, in many ways this field is the motivation for the entire design of this file. Since it is a mixed system we have two separate terms that make it up, the potential and the gradient. These are not derived from one and other, rather they are weakly enforced to agree. This buys us an additional order of convergence in the potential which in tern allows us to use a lower resolution mesh.
- struct IsGlobalScalarTerm

  Helper struct to determine if a type is a global scalar term (GlobalScalarTerm).

- struct IsPhysicalSpatialTerm

  Helper struct to determine if a type is a physical spatial term (PhysicalSpatialTerm).
- struct IsSpatialTerm

  Helper struct to determine if a type is a spatial term (SpatialTerm).
- struct residual_block

  Final class for residual blocks with a fixed index, derived from residual_block_base.
- struct residual_block_base

  Base class for residual blocks, derived from block.
- struct rotation

  The rotational field, represented as the angular velocity term, given that we currently only model rigid rotation this is a global scalar. A future extension which models differential rotation will likely need to modify this to some vector field or other more complex representation of the rotation.
- struct SpatialTermTraits

  Helper struct representing the traits of some spatial term type, including its functional space, support, family, order, and rank.
- struct surface_deformation

  The surface corollary field to the full displacement field. This is in some ways more general than the others in so far as it is not pointwise defined over the domain. Rather, it is some general field of numbers, divorced from the underlying discrimination, that is used to represent the surface deformation of the system. This may be used, for example to represent spherical harmonic coefficients of the surface deformation. For more information see the surface and deformation systems.
- struct term

  The term struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Note that generally you should always use term rather than term_base as term associates to a particular field at compile time.
- struct term_base

  The term_base struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Generally you should always use term rather than term_base as term associates to a particular field at compile time.
- struct TermDescriptor

  Runtime descriptor for a term, containing its associated field and name. Two TermDescriptors are equal if they have identical fields and names.
- struct value_block

  Final class for value blocks with a fixed index, derived from value_block_base.
- struct value_block_base

  Base class for value blocks, derived from block.

Concepts

- concept Field

  Concept for field types in the discretization framework. To be a field a type must derive from the field base class.
- concept Term

  Concept for term types in the discretization framework. To be a term a type must derive from term_base, must have a field_type member, and that field_type member must be a valid field.
- concept HasFunctionalSpace

  Concept for term types that have a functional space. To have a functional space a type must first be a valid Term and must have a FunctionalSpace member type.

- concept HasSupport

  Concept for term types that have a domain over which they are supported. To have support a type must first be a valid Term and must have a Support member type.
- concept HasQuantity

  Concept for term types that have a physical quantity associated with them. To have a quantity a type must first be a valid Term and must have a Quantity member type.
- concept HasValueBlock

  Concept for term types that have a value block associated with them. To have a value block a type must first be a valid Term and must have a value member type.
- concept HasResidualBlock

  Concept for term types that have a residual block associated with them. To have a residual block a type must first be a valid Term and must have a residual member type.
- concept SpatialTerm

  Concept for term types that are spatial terms. A spatial term is a term that has a functional space (HasFunctionalSpace) and a support (HasSupport).
- concept GlobalScalarTerm

  Concept for term types that are global scalar terms. A global scalar term is a term that has support (HasSupport) but does not have a functional space (HasFunctionalSpace) and is derived from the global_scalar class.
- concept PhysicalSpatialTerm

  Concept for a physical spatial term. A physical spatial term is a spatial term (SpatialTerm) that has a physical quantity associated with it (HasQuantity).

Typedefs

- template<Term TermT>
  using FieldOf = std::remove_cvref_t<TermT>::field_type

  Extract the field type associated with a given term type. Will have a type of whatever TermT::field_type is.
- template<HasFunctionalSpace TermT>
  using FunctionalSpaceOf = std::remove_cvref_t<TermT>::FunctionalSpace

  Extract the functional space type associated with a given term type. Will have a type of whatever TermT::FunctionalSpace is.
- template<HasSupport TermT>
  using SupportOf = std::remove_cvref_t<TermT>::Support

  Extract the support type associated with a given term type. Will have a type of whatever TermT::Support is.
- template<HasQuantity TermT>
  using QuantityOf = std::remove_cvref_t<TermT>::Quantity

  Extract the quantity type associated with a given term type. Will have a type of whatever TermT::Quantity is.
- template<typename FieldT>
  using TermsOf = FieldT::Terms

  Extract the terms associated with a given field type. Will have a type of whatever FieldT::Terms is.

Variables

- constexpr int DYNAMIC_BLOCK_SIZE = -1
- template<Field FieldT>
  constexpr FieldDescriptor field_descriptor {.name = FieldT::name}

  Create a FieldDescriptor for a given field type.
- template<Term TermT>
  constexpr TermDescriptor term_descriptor

  Create a TermDescriptor for a given term type.

8.8.1 Typedef Documentation

8.8.1.1 FieldOf

template<Term TermT>
using serif::discretization::blocks::FieldOf = std::remove_cvref_t<TermT>::field_type

Extract the field type associated with a given term type. Will have a type of whatever TermT::field_type is.

Template Parameters

TermT	The term type from which to extract the field type.

8.8.1.2 FunctionalSpaceOf

template<HasFunctionalSpace TermT>
using serif::discretization::blocks::FunctionalSpaceOf = std::remove_cvref_t<TermT>::FunctionalSpace

Extract the functional space type associated with a given term type. Will have a type of whatever TermT↩ ::FunctionalSpace is.

Template Parameters

TermT	The term type from which to extract the functional space type.

Todo For consistency with other types consider renaming TermT::FunctionalSpace to TermT::functional_↩ space. This would require updating all term definitions to use the new naming convention. CLion or some other IDE should be able to handle this refactor easily.

8.8.1.3 QuantityOf

template<HasQuantity TermT>
using serif::discretization::blocks::QuantityOf = std::remove_cvref_t<TermT>::Quantity

Extract the quantity type associated with a given term type. Will have a type of whatever TermT::Quantity is.

Template Parameters

TermT	The term type from which to extract the quantity type.

Todo For consistency with other types consider renaming TermT::Quantity to TermT::quantity. This would require updating all term definitions to use the new naming convention. CLion or some other IDE should be able to handle this refactor easily.

8.8.1.4 SupportOf

template<HasSupport TermT>
using serif::discretization::blocks::SupportOf = std::remove_cvref_t<TermT>::Support

Extract the support type associated with a given term type. Will have a type of whatever TermT::Support is.

Template Parameters

TermT	The term type from which to extract the support type.

Todo For consistency with other types consider renaming TermT::Support to TermT::support. This would require updating all term definitions to use the new naming convention. CLion or some other IDE should be able to handle this refactor easily.

8.8.1.5 TermsOf

template<typename FieldT>
using serif::discretization::blocks::TermsOf = FieldT::Terms

Extract the terms associated with a given field type. Will have a type of whatever FieldT::Terms is.

Template Parameters

FieldT	The field type from which to extract the terms.

Todo For consistency with other types consider renaming FieldT::Terms to FieldT::terms. This would require updating all field definitions to use the new naming convention. CLion or some other IDE should be able to handle this refactor easily.

8.8.2 Variable Documentation

8.8.2.1 DYNAMIC_BLOCK_SIZE

int serif::discretization::blocks::DYNAMIC_BLOCK_SIZE = -1 [inline], [constexpr]

8.8.2.2 field_descriptor

template<Field FieldT>
FieldDescriptor serif::discretization::blocks::field_descriptor {.name = FieldT::name} [inline], [constexpr]

Create a FieldDescriptor for a given field type.

Template Parameters

FieldT	The field type for which to create the descriptor.

8.8.2.3 term_descriptor

template<Term TermT>
TermDescriptor serif::discretization::blocks::term_descriptor [inline], [constexpr]

Initial value:

```
{
    .field = field_descriptor<typename TermT::field_type>,
    .name = TermT::name
}
```

Create a TermDescriptor for a given term type.

Template Parameters

TermT	The term type for which to create the descriptor.

8.9 serif::discretization::domain Namespace Reference

Namespaces

- namespace ids
- namespace mesh
- namespace relation
- namespace schema

Classes

- struct Boundary
- struct BoundaryIDResolver
- struct BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > >
- struct BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > >
- struct BoundarySet
- struct CoreDomain
- struct Domain
- struct DomainIDResolver
- struct DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... > >
- struct DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > >
- class DomainOrSet
- struct DomainOrSetList
- struct DomainSet
- struct EnvelopeDomain
- struct InfinitySurfaceBoundary
- struct StellarSurfaceBoundary
- struct VacuumDomain

Concepts

- concept IsDomain
- concept IsBoundary
- concept IsDomainSet
- concept IsDomainOrSet
- concept DomainIsIdentifiedBy
- concept BoundaryIsIdentifiedBy

Typedefs

- using StellarDomains = DomainSet<CoreDomain, EnvelopeDomain>
- using AllDomains = DomainSet<CoreDomain, EnvelopeDomain, VacuumDomain>

Functions

- template<typename... Ts>
  consteval std::size_t CountVariadicArguments () noexcept

Variables

- template<typename T>
  constexpr bool is_domain_set_v = false
- template<IsDomain... DomainTs>
  constexpr bool is_domain_set_v< DomainSet< DomainTs... > > = true
- template<class Child, class Parent>
  constexpr bool AIsBaseClassOfB

8.9.1 Typedef Documentation

8.9.1.1 AllDomains

using serif::discretization::domain::AllDomains = DomainSet<CoreDomain, EnvelopeDomain, VacuumDomain>

8.9.1.2 StellarDomains

using serif::discretization::domain::StellarDomains = DomainSet<CoreDomain, EnvelopeDomain>

8.9.2 Function Documentation

8.9.2.1 CountVariadicArguments()

template<typename... Ts>
std::size_t serif::utils::misc::std_helper::CountVariadicArguments () [nodiscard], [consteval], [noexcept]

8.9.3 Variable Documentation

8.9.3.1 AIsBaseClassOfB

template<class Child, class Parent>
bool serif::utils::misc::std_helper::AIsBaseClassOfB [constexpr]

8.9.3.2 is_domain_set_v

template<typename T>
bool serif::discretization::domain::is_domain_set_v = false [constexpr]

8.9.3.3 is_domain_set_v< DomainSet< DomainTs... > >

template<IsDomain... DomainTs>
bool serif::discretization::domain::is_domain_set_v< DomainSet< DomainTs... > > = true [constexpr]

8.10 serif::discretization::domain::ids Namespace Reference

Namespaces

- namespace lists
- namespace validation

Classes

- struct BoundaryDescriptor

  The BoundaryDescriptor struct is a runtime representation of a boundary, containing its name and integer attribute ID. This is used to project compile-time BoundaryID types into runtime values.
- struct BoundaryID

  The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation.
- struct DomainDescriptor

  The DomainDescriptor struct is a runtime representation of a domain, containing its name and integer attribute ID. This is used to project compile-time DomainID types into runtime values.
- struct DomainID

  The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation.

Concepts

- concept IsDomainID

  Concept for types that are domain IDs in the discretization framework. To be a domain ID a type must satisfy the is_domain_id_v trait.
- concept IsBoundaryID

  Concept for types that are boundary IDs in the discretization framework. To be a boundary ID a type must satisfy the is_boundary_id_v trait.
- concept IsDomainOrBoundaryID

  Concept for types that are either domain IDs or boundary IDs in the discretization framework.

Variables

- template<typename T>
  constexpr bool is_domain_id_v = false

  Trait to determine if a type is a domain ID. Defaults to false for all types.
- template<IsDomain DomainT, int IdentifierValue>
  constexpr bool is_domain_id_v< DomainID< DomainT, IdentifierValue > > = true

  Trait to determine if a type is a domain ID. Specialization which is true for valid DomainID types.
- template<typename T>
  constexpr bool is_boundary_id_v = false

  Trait to determine if a type is a boundary ID. Defaults to false for all types.
- template<IsBoundary BoundaryT, int IdentifierValue>
  constexpr bool is_boundary_id_v< BoundaryID< BoundaryT, IdentifierValue > > = true

  Trait to determine if a type is a boundary ID. Specialization which is true for valid BoundaryID types.

8.10.1 Variable Documentation

8.10.1.1 is_boundary_id_v

template<typename T>
bool serif::discretization::domain::ids::is_boundary_id_v = false [constexpr]

Trait to determine if a type is a boundary ID. Defaults to false for all types.

Template Parameters

T	The type to check.

8.10.1.2 is_boundary_id_v< BoundaryID< BoundaryT, IdentifierValue > >

template<IsBoundary BoundaryT, int IdentifierValue>
bool serif::discretization::domain::ids::is_boundary_id_v< BoundaryID< BoundaryT, IdentifierValue > > = true [constexpr]

Trait to determine if a type is a boundary ID. Specialization which is true for valid BoundaryID types.

Template Parameters

T	The type to check.

8.10.1.3 is_domain_id_v

template<typename T>
bool serif::discretization::domain::ids::is_domain_id_v = false [constexpr]

Trait to determine if a type is a domain ID. Defaults to false for all types.

Template Parameters

T	The type to check.

8.10.1.4 is_domain_id_v< DomainID< DomainT, IdentifierValue > >

template<IsDomain DomainT, int IdentifierValue>
bool serif::discretization::domain::ids::is_domain_id_v< DomainID< DomainT, IdentifierValue > > = true [constexpr]

Trait to determine if a type is a domain ID. Specialization which is true for valid DomainID types.

Template Parameters

T	The type to check.

8.11 serif::discretization::domain::ids::lists Namespace Reference

Classes

- struct BoundaryIDList

  A list of boundary IDs in the discretization framework.
- struct DomainIDList

  A list of domain IDs in the discretization framework.

Concepts

- concept IsDomainIDList

  Concept for types that are domain ID lists in the discretization framework. To be a domain ID list a type must satisfy the is_domain_id_list_v trait.
- concept IsBoundaryIDList

  Concept for types that are boundary ID lists in the discretization framework. To be a boundary ID list a type must satisfy the is_boundary_id_list_v trait.

Functions

- template<typename... Ts>
  consteval std::size_t CountVariadicArguments () noexcept

Variables

- template<typename T>
  constexpr bool is_domain_id_list_v = false

  Trait to determine if a type is a domain ID list. Defaults to false for all types.
- template<IsDomainID... DomainIDTs>
  constexpr bool is_domain_id_list_v< DomainIDList< DomainIDTs... > > = true

  Trait to determine if a type is a boundary ID list. Specialization which is true for valid BoundaryIDList types.
- template<typename T>
  constexpr bool is_boundary_id_list_v = false

  Trait to determine if a type is a boundary ID list. Defaults to false for all types.
- template<IsBoundaryID... BoundaryIDTs>
  constexpr bool is_boundary_id_list_v< BoundaryIDList< BoundaryIDTs... > > = true

  Trait to determine if a type is a boundary ID list. Specialization which is true for valid BoundaryIDList types.

8.11.1 Function Documentation

8.11.1.1 CountVariadicArguments()

template<typename... Ts>
std::size_t serif::utils::misc::std_helper::CountVariadicArguments () [nodiscard], [consteval], [noexcept]

8.11.2 Variable Documentation

8.11.2.1 is_boundary_id_list_v

template<typename T>
bool serif::discretization::domain::ids::lists::is_boundary_id_list_v = false [constexpr]

Trait to determine if a type is a boundary ID list. Defaults to false for all types.

Template Parameters

T	The type to check.

8.11.2.2 is_boundary_id_list_v< BoundaryIDList< BoundaryIDTs... > >

template<IsBoundaryID... BoundaryIDTs>
bool serif::discretization::domain::ids::lists::is_boundary_id_list_v< BoundaryIDList< BoundaryIDTs... > > = true [constexpr]

Trait to determine if a type is a boundary ID list. Specialization which is true for valid BoundaryIDList types.

Template Parameters

T	The type to check.

8.11.2.3 is_domain_id_list_v

template<typename T>
bool serif::discretization::domain::ids::lists::is_domain_id_list_v = false [constexpr]

Trait to determine if a type is a domain ID list. Defaults to false for all types.

Template Parameters

T	The type to check.

8.11.2.4 is_domain_id_list_v< DomainIDList< DomainIDTs... > >

template<IsDomainID... DomainIDTs>
bool serif::discretization::domain::ids::lists::is_domain_id_list_v< DomainIDList< DomainIDTs... > > = true [constexpr]

Trait to determine if a type is a boundary ID list. Specialization which is true for valid BoundaryIDList types.

Template Parameters

T	The type to check.

8.12 serif::discretization::domain::ids::validation Namespace Reference

Concepts

- concept HaveUniqueDomainIDs

  Concept to check if a pack of domain IDs have unique ID values. This is used to ensure that no two domain IDs in a list have the same ID.
- concept HaveUniqueBoundaryIDs

  Concept to check if a pack of boundary IDs have unique ID values. This is used to ensure that no two boundary IDs in a list have the same ID.
- concept HaveUniqueDomains

  Concept to check if a pack of domain IDs have unique domain types. This is used to ensure that no two domain IDs in a list refer to the same domain type.
- concept HaveUniqueBoundaries

  Concept to check if a pack of boundary IDs have unique boundary types. This is used to ensure that no two boundary IDs in a list refer to the same boundary type.
- concept AreDistinctDomainIDs

  Concept to check if a pack of domain IDs are distinct. This means that each ID is a valid domain ID, no two IDs have the same integer value, and no two IDs refer to the same domain type.
- concept AreDistinctBoundaryIDs

  Concept to check if a pack of boundary IDs are distinct. This means that each ID is a valid boundary ID, no two IDs have the same integer value, and no two IDs refer to the same boundary type.

Functions

- template<typename... Ts>
  requires (std::equality_comparable<Ts> && ...)
  consteval bool all_unique_values (Ts... values) noexcept

  Utility function to check if a pack of values are all unique.

8.12.1 Function Documentation

8.12.1.1 all_unique_values()

template<typename... Ts>
requires (std::equality_comparable<Ts> && ...)
bool serif::discretization::domain::ids::validation::all_unique_values (
Ts... values) [nodiscard], [consteval], [noexcept]

Utility function to check if a pack of values are all unique.

Template Parameters

Ts	The types of the values to check.

Parameters

values	The values to check for uniqueness.

Returns

true if all values are unique, false otherwise.

Todo Really this should probably be refactored into the utils module since this is not super specific to domainIDs.

8.13 serif::discretization::domain::mesh Namespace Reference

Classes

- struct FaceElements
- class MeshTopology

8.14 serif::discretization::domain::relation Namespace Reference

Namespaces

- namespace lists
- namespace validation

Classes

- struct DomainBoundary
- struct DomainRelation
- struct FullyConnected
- struct Inscribed

Concepts

- concept IsRelation

Functions

- template<typename... Ts>
  consteval std::size_t CountVariadicArguments () noexcept

Variables

- template<class Child, class Parent>
  constexpr bool AIsBaseClassOfB

8.14.1 Function Documentation

8.14.1.1 CountVariadicArguments()

template<typename... Ts>
std::size_t serif::utils::misc::std_helper::CountVariadicArguments () [nodiscard], [consteval], [noexcept]

8.14.2 Variable Documentation

8.14.2.1 AIsBaseClassOfB

template<class Child, class Parent>
bool serif::utils::misc::std_helper::AIsBaseClassOfB [constexpr]

8.15 serif::discretization::domain::relation::lists Namespace Reference

Classes

- struct RelationList

Concepts

- concept IsRelationList

Functions

- template<typename... Ts>
  consteval std::size_t CountVariadicArguments () noexcept

Variables

- template<typename T>
  constexpr bool is_relation_list_v = false
- template<IsRelation... RelationTs>
  constexpr bool is_relation_list_v< RelationList< RelationTs... > > = true

8.15.1 Function Documentation

8.15.1.1 CountVariadicArguments()

template<typename... Ts>
std::size_t serif::utils::misc::std_helper::CountVariadicArguments () [nodiscard], [consteval], [noexcept]

8.15.2 Variable Documentation

8.15.2.1 is_relation_list_v

template<typename T>
bool serif::discretization::domain::relation::lists::is_relation_list_v = false [constexpr]

8.15.2.2 is_relation_list_v< RelationList< RelationTs... > >

template<IsRelation... RelationTs>
bool serif::discretization::domain::relation::lists::is_relation_list_v< RelationList< RelationTs... > > = true [constexpr]

8.16 serif::discretization::domain::relation::validation Namespace Reference

Classes

- struct RelationsUseRegisteredEntities
- struct RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, lists::RelationList< RelationTs... > >
- struct RelationUsesRegisteredEntities
- struct RelationUsesRegisteredEntities< DomainBoundary< BoundaryT, DomainTs... >, DomainIDsT, BoundaryIDsT
- struct RelationUsesRegisteredEntities< FullyConnected< DomainT >, DomainIDsT, BoundaryIDsT >
- struct RelationUsesRegisteredEntities< Inscribed< InnerDomainT, OuterDomainT >, DomainIDsT, BoundaryIDsT >
- struct RelationValidationResult

Concepts

- concept HaveValidRelationEntities

Enumerations

- enum class RelationValidationFailure : std::uint8_t {
  None , DomainAbsent , DomainDisconnected , InnerDomainAbsent ,
  OuterDomainAbsent , InnerDomainHasNoBoundary , InnerDomainTouchesMeshBoundary ,
  InnerDomainTouchesUnexpectedDomain ,
  DomainBoundaryAbsent , DomainBoundaryTaggedFaceHasWrongTopology , DomainBoundaryTaggedFaceTouchesUne
  , DomainBoundaryExpectedFaceIsUntagged ,
  DomainBoundaryExpectedFaceHasWrongID }

8.16.1 Enumeration Type Documentation

8.16.1.1 RelationValidationFailure

enum class serif::discretization::domain::relation::validation::RelationValidationFailure : std::uint8_t [strong]

Enumerator

None	
DomainAbsent	
DomainDisconnected	
InnerDomainAbsent	
OuterDomainAbsent	
InnerDomainHasNoBoundary	
InnerDomainTouchesMeshBoundary	
InnerDomainTouchesUnexpectedDomain	
DomainBoundaryAbsent	
DomainBoundaryTaggedFaceHasWrongTopology	
DomainBoundaryTaggedFaceTouchesUnexpectedDomain	
DomainBoundaryExpectedFaceIsUntagged	
DomainBoundaryExpectedFaceHasWrongID	

8.17 serif::discretization::domain::schema Namespace Reference

Namespaces

- namespace validation

Classes

- struct BoundaryID

  The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation.
- struct BoundaryIDList

  A list of boundary IDs in the discretization framework.
- struct DomainID

  The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation.
- struct DomainIDList

  A list of domain IDs in the discretization framework.
- struct DomainSchema
- struct RelationList

Concepts

- concept IsSchema

Typedefs

- using CoreEnvelopeVacuumDomainSchema

Functions

- template<IsDomainOrSet DomainT, IsSchema SchemaT>
  mfem::Array< int > make_attribute_marker (const mfem::Mesh &mesh)

Variables

- template<typename S>
  constexpr bool is_schema_v = false
- template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
  constexpr bool is_schema_v< DomainSchema< DomainIDs, BoundaryIDs, Relations > > = true

8.17.1 Typedef Documentation

8.17.1.1 CoreEnvelopeVacuumDomainSchema

using serif::discretization::domain::schema::CoreEnvelopeVacuumDomainSchema

Initial value:

```
DomainSchema<
    DomainIDList<
        DomainID<CoreDomain, 1>,
        DomainID<EnvelopeDomain, 2>,
        DomainID<VacuumDomain, 3>
    >,
    BoundaryIDList<
        BoundaryID<StellarSurfaceBoundary, 1>,
        BoundaryID<InfinitySurfaceBoundary, 2>
    >,
    RelationList<

        relation::FullyConnected<CoreDomain>,
        relation::FullyConnected<EnvelopeDomain>,
        relation::FullyConnected<VacuumDomain>,



        relation::Inscribed<CoreDomain, EnvelopeDomain>,
        relation::Inscribed<StellarDomains, VacuumDomain>,



        relation::DomainBoundary<StellarSurfaceBoundary, StellarDomains, VacuumDomain>,
        relation::DomainBoundary<InfinitySurfaceBoundary, VacuumDomain>
    >
>
```

8.17.2 Function Documentation

8.17.2.1 make_attribute_marker()

template<IsDomainOrSet DomainT, IsSchema SchemaT>
mfem::Array< int > serif::discretization::domain::schema::make_attribute_marker (
const mfem::Mesh & mesh) [nodiscard]

8.17.3 Variable Documentation

8.17.3.1 is_schema_v

template<typename S>
bool serif::discretization::domain::schema::is_schema_v = false [constexpr]

8.17.3.2 is_schema_v< DomainSchema< DomainIDs, BoundaryIDs, Relations > >

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
bool serif::discretization::domain::schema::is_schema_v< DomainSchema< DomainIDs, BoundaryIDs, Relations > > = true [constexpr]

8.18 serif::discretization::domain::schema::validation Namespace Reference

Classes

- struct DomainBoundary
- struct FullyConnected
- struct Inscribed
- struct RelationValidationResult
- struct RelationValidator
- struct RelationValidator< DomainBoundary< BoundaryT, DomainTs... > >
- struct RelationValidator< FullyConnected< DomainT > >
- struct RelationValidator< Inscribed< InnerT, OuterT > >
- struct SchemaRelationValidationResult
- struct SchemaRelationValidator
- struct SchemaRelationValidator< SchemaT, relation::lists::RelationList< RelationTs... > >
- struct SchemaValidationResult

Enumerations

- enum class RelationValidationFailure
- enum class RelationValidationFailure
- enum class RelationValidationFailure

Functions

- template<IsSchema SchemaT>
  SchemaValidationResult validate_schema (const mesh::MeshTopology &topology)
- template<IsSchema SchemaT>
  SchemaValidationResult validate_schema (const mfem::Mesh &mfemMesh)

8.18.1 Enumeration Type Documentation

8.18.1.1 RelationValidationFailure [1/3]

enum class serif::discretization::domain::relation::validation::RelationValidationFailure : std::uint8_t [strong]

8.18.1.2 RelationValidationFailure [2/3]

enum class serif::discretization::domain::relation::validation::RelationValidationFailure : std::uint8_t [strong]

8.18.1.3 RelationValidationFailure [3/3]

enum class serif::discretization::domain::relation::validation::RelationValidationFailure : std::uint8_t [strong]

8.18.2 Function Documentation

8.18.2.1 validate_schema() [1/2]

template<IsSchema SchemaT>
SchemaValidationResult serif::discretization::domain::schema::validation::validate_schema (
const mesh::MeshTopology & topology) [nodiscard]

8.18.2.2 validate_schema() [2/2]

template<IsSchema SchemaT>
SchemaValidationResult serif::discretization::domain::schema::validation::validate_schema (
const mfem::Mesh & mfemMesh) [nodiscard]

8.19 serif::discretization::forms Namespace Reference

Classes

- struct boundary_flux

  The boundary_flux form represents the boundary flux operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the boundary flux operation.
- struct divergence

  The divergence form represents the divergence operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the divergence operation.
- struct force

  The force form represents the force term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the force operation.
- struct form

  The form struct is a tag type that is used to identify a form in the discretization. It is used to group together forms that belong to the same mathematical operation. Each form should have a unique name and a list of operand types that belong to it.
- struct FormDescriptor

  Runtime form descriptor including runtime term information. Used to represented the compile-time form type at runtime.
- struct FormTraits

  Compile-time traits for forms, used to extract types and properties from template parameters. This struct is specialized for each form type to provide specific traits such as trial and test order reductions.
- struct FormTraits< boundary_flux< TrialT, TestT > >

  Specialization of FormTraits for the boundary_flux form, which has no order reduction for either trial or test terms.
- struct FormTraits< divergence< TrialT, TestT > >

  Specialization of FormTraits for the divergence form, which has an order reduction of 1, to satisfy the Inf-Sup condition, for the trial term and no order reduction for the test term.
- struct FormTraits< force< TrialT, TestT > >

  Specialization of FormTraits for the force form, which has no order reduction for either trial or test terms.
- struct FormTraits< mass< TrialT, TestT > >

Specialization of FormTraits for the mass form, which has no order reduction for either trial or test terms.

- struct FormTraits< projection< TrialT, TestT > >

  Specialization of FormTraits for the projection form, which has no order reduction for either trial or test terms.

- struct FormTraits< source< TrialT, TestT > >

  Specialization of FormTraits for the source form, which has no order reduction for either trial or test terms.

- struct mass

  The mass form represents the mass matrix operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the mass operation.

- struct Operand

  Compile time representation of some operand. An operand associates a term to an operation type (e.g. the gradient of gravitational potential where gravitational potential is the term and gradient is the operation).

- struct OperandResult

  General case for compile time metadata associating some operand to the value type of its result. These use the OperationValue struct and the information in the operand to determine the value type of the result of that operand.

- struct OperandResult< OperandT >

  Specialization of the OperandResult struct for operands that are valid operands (satisfy the IsOperand concept) and are spatial terms (e.g defined over the domain, think mass density). This specialization uses the OperationValue struct to determine the value type of the result of the operand based on its term type and operation type.

- struct Operation

  The Operation struct defines a set of operations that can be applied to terms in the context of a form. These operations include value, gradient, divergence, curl, and normal trace. Similar to the statement I made at the top of forms.hpp, if you are a developer looking to extend the mathematical capabilities of SERiF this is a place you should be looking early on.

- struct OperationValue

  General case for compile time metadata associating a tensor rank to an operation. The specialization of this will also specific what the return value of an operation of that rank will be.

- struct OperationValue< 0, Operation::Gradient >

  The gradient of a rank 0 tensor is a rank 1 tensor (a vector of doubles)

- struct OperationValue< 0, Operation::Value >

  The value of a rank 0 tensor is a scalar, a double in the case of SERiF where we restrict ourselves to double precision.

- struct OperationValue< 1, Operation::Curl >

  The curl of a rank 1 tensor is a rank 1 tensor (a vector of doubles)

- struct OperationValue< 1, Operation::Divergence >

  The divergence of a rank one tensor is a scalar (a double)

- struct OperationValue< 1, Operation::NormalTrace >
- struct OperationValue< 1, Operation::Value >

  The value of a rank 1 tensor is a rank 1 tensor (a vector of doubles in SERiF)

- struct projection

  The projection form represents the projection operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the projection operation.

- struct source

  The source form represents the source term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the source operation.

Concepts

- concept Form

  Concept for types that are forms in the discretization framework. In order satisfy this concept a type must derive from the form base class and must define a member type called operand_types which is intended to be a type list of the trial and test terms for the form.
- concept IsOperation
- concept IsOperand

  Concept to validate a valid operand. An operand is valid if the following conditions are satisfied.

Typedefs

- using VectorValue = std::array<double, 3>

  Alias for a rank 1 tensor of dimension 3 with elements doubles.
- using MatrixValue = std::array<VectorValue, 3>

  Alias for a rank 2 tensor of dimension 3 with cells as doubles.
- template<IsOperand OperandT>
  using OperandValue = OperandResult<OperandT>::type

  Alias for the value type of an operand. This is a convenience alias that allows you to easily access the value type of an operand without having to explicitly use the OperandResult struct.
- template<typename OutputT, typename InputT>
  using PartialDerivative = std::conditional_t<std::same_as<OutputT, double>, InputT, std::conditional_t<std::same_as<InputT, double>, OutputT, MatrixValue>>

  Alias for the partial derivative of some output type with respect to some input type. This is a convenience alias that allows you to easily access the partial derivative type without having to explicitly use the OperandResult struct.

Functions

- template<typename ValueT>
  decltype(auto) component (ValueT &value, int index)

  Read one entry of a scalar/vector/matrix value. This is a convenience function that allows you to easily access the components of a value type without having to explicitly use the OperandResult struct.
- template<typename OutputT, typename InputT>
  double derivative_component (const PartialDerivative< OutputT, InputT > &value, int output, int input)

  Read one entry of a scalar/vector/matrix local derivative.

Variables

- template<Form FormT>
  constexpr FormDescriptor form_descriptor

  Specialization of the FormDescriptor class at compile time for some particular compile-time form. This allows us to generate constexpr instances of this struct at compile time and usable at runtime.
- template<typename ValueT>
  constexpr int component_count = std::same_as<ValueT, double> ? 1 : 3

  Helper variable template to determine the number of components in the value type of an operand. This is a convenience variable that allows you to easily access the number of components in the value type of an operand without having to explicitly use the OperandResult struct.

8.19.1 Typedef Documentation

8.19.1.1 MatrixValue

using serif::discretization::forms::MatrixValue = std::array<VectorValue, 3>

Alias for a rank 2 tensor of dimension 3 with cells as doubles.

8.19.1.2 OperandValue

template<IsOperand OperandT>
using serif::discretization::forms::OperandValue = OperandResult<OperandT>::type

Alias for the value type of an operand. This is a convenience alias that allows you to easily access the value type of an operand without having to explicitly use the OperandResult struct.

Template Parameters

OperandT	The operand type to determine the value type for.

8.19.1.3 PartialDerivative

template<typename OutputT, typename InputT>
using serif::discretization::forms::PartialDerivative = std::conditional_t<std::same_as<OutputT, double>, InputT, std::conditional_t<std::same_as<InputT, double>, OutputT, MatrixValue>>

Alias for the partial derivative of some output type with respect to some input type. This is a convenience alias that allows you to easily access the partial derivative type without having to explicitly use the OperandResult struct.

Template Parameters

OutputT	The output type of the partial derivative.
InputT	The input type of the partial derivative.

Todo Should probably be consolidated with the other PartialDerivative system... No, should definitely be consolidated...

8.19.1.4 VectorValue

using serif::discretization::forms::VectorValue = std::array<double, 3>

Alias for a rank 1 tensor of dimension 3 with elements doubles.

Todo This file was a mistake, the information it is is helpful yes, however, it has lost generality, lost good type information, and does not need to be separate Really we should refactor this to be part of Operation, remove the assumption of 3 dimensionality, and probably fall back to something like a span rather than an array.

8.19.2 Function Documentation

8.19.2.1 component()

```
template<typename ValueT>
decltype(auto) serif::discretization::forms::component (
                ValueT & value,
                int index)
```

Read one entry of a scalar/vector/matrix value. This is a convenience function that allows you to easily access the components of a value type without having to explicitly use the OperandResult struct.

Template Parameters

ValueT	The value type to read from.

Parameters

value	The value to read from.
index	The index of the component to read.

Returns

The component of the value at the specified index.

8.19.2.2 derivative_component()

```
template<typename OutputT, typename InputT>
double serif::discretization::forms::derivative_component (
                const PartialDerivative< OutputT, InputT > & value,
                int output,
                int input)
```

Read one entry of a scalar/vector/matrix local derivative.

Template Parameters

OutputT	The output type of the partial derivative.
InputT	The input type of the partial derivative.

Parameters

value	The partial derivative to read from.
output	The index of the output component to read.
input	The index of the input component to read.

Returns

The component of the partial derivative at the specified indices.

8.19.3 Variable Documentation

8.19.3.1 component_count

template<typename ValueT>
int serif::discretization::forms::component_count = std::same_as<ValueT, double> ? 1 : 3 [inline], [constexpr]

Helper variable template to determine the number of components in the value type of an operand. This is a convenience variable that allows you to easily access the number of components in the value type of an operand without having to explicitly use the OperandResult struct.

Template Parameters

OperandT	The operand type to determine the number of components for.

Todo This introduces an assumption of three-dimensionality, we say "this is either a scalar or a tensor of dimension 3". We should probably relax this assumption.

8.19.3.2 form_descriptor

template<Form FormT>
FormDescriptor serif::discretization::forms::form_descriptor [inline], [constexpr]

Initial value:

```
{
    .name = FormT::name,
    .trial = blocks::term_descriptor<typename FormT::trial_type>,
    .test = blocks::term_descriptor<typename FormT::test_type>
}
```

Specialization of the FormDescriptor class at compile time for some particular compile-time form. This allows us to generate constexpr instances of this struct at compile time and usable at runtime.

Template Parameters

FormT	The form to generate a descriptor for

8.20 serif::discretization::functional_space Namespace Reference

Classes

- struct DiscreteFunctionalSpace

  Compile-time descriptor for a discrete functional space. The difference between a FunctionalSpace and a discrete functional space is that a discrete functional space carries some additional information which is needed to actually construct a finite element space (e.g. polynomial order, and rank). Once you have a discrete functional space you have, for MFEM at least, enough information to fully define a finite element space.
- struct Rank

  Compile-time descriptor for the rank of a functional space.
- struct SpaceDescriptor

  Compile-time descriptor for a functional space. This struct is used to describe the family, continuity, trace, and mapping of a functional space at compile time.

Concepts

- concept IsSpaceDescriptor

  Concept to validate a valid space descriptor. A space descriptor is valid if it is of type SpaceDescriptor. This is a very simple concept that just checks that the type is exactly SpaceDescriptor.

Enumerations

- enum class Family { l2 , h1 , hdiv , hcurl }

  Compile-time descriptor for the family, continuity, trace, and mapping of a functional space.
- enum class Continuity { discontinuous , continuous , normal_continuous , tangential_continuous }

  Compile-time descriptor for the continuity of a functional space. This enum is parallel to the Family enum.
- enum class Trace { none , value , normal , tangential }

  Compile-time descriptor for the trace of a functional space. This enum is parallel to the Family enum.
- enum class Mapping { identity , contravariant_piola , covariant_piola }

  Compile-time descriptor for the mapping of a functional space. This tells the DomainMapper which piola transformations to use when moving between reference and physical spaces.

Variables

- static auto Family2StringMap = meta::enum_to_enum_string_map<Family>()

  Mapping from Family enum values to their string representations.
- static auto String2FamilyMap = meta::enum_to_string_enum_map<Family>()

  Mapping from string representations to Family enum values.
- static auto Continuity2StringMap = meta::enum_to_enum_string_map<Continuity>()

  Mapping from Continuity enum values to their string representations.
- static auto String2ContinuityMap = meta::enum_to_string_enum_map<Continuity>()

  Mapping from string representations to Continuity enum values.
- static auto Trace2StringMap = meta::enum_to_enum_string_map<Trace>()

  Mapping from Trace enum values to their string representations.
- static auto String2TraceMap = meta::enum_to_string_enum_map<Trace>()

  Mapping from string representations to Trace enum values.
- static auto Mapping2StringMap = meta::enum_to_enum_string_map<Mapping>()

  Mapping from Mapping enum values to their string representations.
- static auto String2MappingMap = meta::enum_to_string_enum_map<Mapping>()

  Mapping from string representations to Mapping enum values.
- static constexpr SpaceDescriptor L2

  Compile-time descriptor for an L2 functional space. This is a convenience descriptor that allows you to easily create an L2 functional space descriptor without having to specify all the parameters. The specifications are.
- static constexpr SpaceDescriptor H1

  Compile-time descriptor for an H1 functional space. This is a convenience descriptor that allows you to easily create an H1 functional space descriptor without having to specify all the parameters. The specifications are.
- static constexpr SpaceDescriptor Hdiv

  Compile-time descriptor for an H(div) functional space. This is a convenience descriptor that allows you to easily create an H(div) functional space descriptor without having to specify all the parameters. The specifications are.
- static constexpr SpaceDescriptor Hcurl

  Compile-time descriptor for an H(curl) functional space. This is a convenience descriptor that allows you to easily create an H(curl) functional space descriptor without having to specify all the parameters. The specifications are.

8.20.1 Enumeration Type Documentation

8.20.1.1 Continuity

enum class serif::discretization::functional_space::Continuity [strong]

Compile-time descriptor for the continuity of a functional space. This enum is parallel to the Family enum.

Todo There may be a better way to store these relations that forces them to be parallel rather than just relying on the programmer to keep them in sync.

Enumerator

discontinuous	Discontinuous functions, typically used for L2 spaces.
continuous	C0 continuous functions, typically used for H1 spaces.
normal_continuous	Functions with normal continuity, typically used for H(div) spaces.
tangential_continuous	Functions with tangential continuity, typically used for H(curl) spaces.

8.20.1.2 Family

enum class serif::discretization::functional_space::Family [strong]

Compile-time descriptor for the family, continuity, trace, and mapping of a functional space.

Note

These are almost all of the functional spaces that MFEM supports, if in the future MFEM adds some other space and you wish to extend SERiF to support it (or you introduce some non MFEM backend with another space) then one early place you will need to modify is this enum.

We do not support mfem::ceed::ParAlgebraicCoarseSpace

Enumerator

l2	L2 space, typically used for discontinuous functions.
h1	H1 space, typically used for continuous functions.
hdiv	H(div) space, typically used for vector fields with normal continuity.
hcurl	H(curl) space, typically used for vector fields with tangential continuity.

8.20.1.3 Mapping

enum class serif::discretization::functional_space::Mapping [strong]

Compile-time descriptor for the mapping of a functional space. This tells the DomainMapper which piola transformations to use when moving between reference and physical spaces.

Enumerator

identity	Identity mapping, typically used for L2 and H1 spaces.
contravariant_piola	Contravariant Piola mapping, preserving the normal trace, typically used for H(div) spaces. ($\frac{1}{\det(J)} J\hat{\phi}(F^{-1}(x))$ transforming some vector field $\phi(x)$ for some transformation F with Jacobian J)
covariant_piola	Covariant Piola mapping, preserving the tangential trace, typically used for H(curl) spaces. ($J^{-T}\hat{\phi}(F^{-1}(x))$ transforming some vector field $\phi(x)$ for some transformation F with Jacobian J)

8.20.1.4 Trace

enum class serif::discretization::functional_space::Trace [strong]

Compile-time descriptor for the trace of a functional space. This enum is parallel to the Family enum.

Enumerator

none	No trace, typically used for L2 spaces.
value	Value trace, typically used for H1 spaces.
normal	Normal trace, typically used for H(div) spaces.
tangential	Tangential trace, typically used for H(curl) spaces.

8.20.2 Variable Documentation

8.20.2.1 Continuity2StringMap

auto serif::discretization::functional_space::Continuity2StringMap = meta::enum_to_enum_string_map<Continuity>() [inline], [static]

Mapping from Continuity enum values to their string representations.

8.20.2.2 Family2StringMap

auto serif::discretization::functional_space::Family2StringMap = meta::enum_to_enum_string_map<Family>() [inline], [static]

Mapping from Family enum values to their string representations.

8.20.2.3 H1

SpaceDescriptor serif::discretization::functional_space::H1 [inline], [static], [constexpr]

Initial value:

```
{
        .family = Family::h1,
        .continuity = Continuity::continuous,
        .trace = Trace::value,
        .mapping = Mapping::identity
    }
```

Compile-time descriptor for an H1 functional space. This is a convenience descriptor that allows you to easily create an H1 functional space descriptor without having to specify all the parameters. The specifications are.

- family = Family::h1
- continuity = Continuity::continuous
- trace = Trace::value
- mapping = Mapping::identity

8.20.2.4 Hcurl

SpaceDescriptor serif::discretization::functional_space::Hcurl [inline], [static], [constexpr]

Initial value:

```
{
        .family = Family::hcurl,
        .continuity = Continuity::tangential_continuous,
        .trace = Trace::tangential,
        .mapping = Mapping::covariant_piola
    }
```

Compile-time descriptor for an H(curl) functional space. This is a convenience descriptor that allows you to easily create an H(curl) functional space descriptor without having to specify all the parameters. The specifications are.

- family = Family::hcurl
- continuity = Continuity::tangential_continuous
- trace = Trace::tangential
- mapping = Mapping::covariant_piola

8.20.2.5 Hdiv

SpaceDescriptor serif::discretization::functional_space::Hdiv [inline], [static], [constexpr]

Initial value:

```
{
    .family = Family::hdiv,
    .continuity = Continuity::normal_continuous,
    .trace = Trace::normal,
    .mapping = Mapping::contravariant_piola
}
```

Compile-time descriptor for an H(div) functional space. This is a convenience descriptor that allows you to easily create an H(div) functional space descriptor without having to specify all the parameters. The specifications are.

- family = Family::hdiv
- continuity = Continuity::normal_continuous
- trace = Trace::normal
- mapping = Mapping::contravariant_piola

8.20.2.6 L2

SpaceDescriptor serif::discretization::functional_space::L2 [inline], [static], [constexpr]

Initial value:

```
{
    .family = Family::l2,
    .continuity = Continuity::discontinuous,
    .trace = Trace::none,
    .mapping = Mapping::identity
}
```

Compile-time descriptor for an L2 functional space. This is a convenience descriptor that allows you to easily create an L2 functional space descriptor without having to specify all the parameters. The specifications are.

- family = Family::l2
- continuity = Continuity::discontinuous
- trace = Trace::none
- mapping = Mapping::identity

8.20.2.7 Mapping2StringMap

auto serif::discretization::functional_space::Mapping2StringMap = meta::enum_to_enum_string_map<Mapping>() [inline], [static]

Mapping from Mapping enum values to their string representations.

8.20.2.8 String2ContinuityMap

auto serif::discretization::functional_space::String2ContinuityMap = meta::enum_to_string_enum_map<Continuity>() [inline], [static]

Mapping from string representations to Continuity enum values.

8.20.2.9 String2FamilyMap

auto serif::discretization::functional_space::String2FamilyMap = meta::enum_to_string_enum_map<Family>() [inline], [static]

Mapping from string representations to Family enum values.

8.20.2.10 String2MappingMap

auto serif::discretization::functional_space::String2MappingMap = meta::enum_to_string_enum_map<Mapping>() [inline], [static]

Mapping from string representations to Mapping enum values.

8.20.2.11 String2TraceMap

auto serif::discretization::functional_space::String2TraceMap = meta::enum_to_string_enum_map<Trace>() [inline], [static]

Mapping from string representations to Trace enum values.

8.20.2.12 Trace2StringMap

auto serif::discretization::functional_space::Trace2StringMap = meta::enum_to_enum_string_map<Trace>() [inline], [static]

Mapping from Trace enum values to their string representations.

8.21 serif::discretization::quadrature Namespace Reference

Namespaces

- namespace backend

Classes

- struct FormRule
- class Policy
- struct QuadratureManifestOptions
- struct QuadratureOptions
- struct QuadratureRoleOptions
- struct QuadratureTermOptions
- struct QuadratureValidationOptions
- struct Query
- struct Resolution
- struct RoleControls
- struct RuleControl
- class RuleSet

Enumerations

- enum class QuadratureRole { discretization , preconditioner , diagnostic , projection }
- enum class MappingKind { none , affine , general , kelvin }
- enum class Mode { fast , production , reference , convergence }

Functions

- template<forms::Form FormT>
  constexpr Query make_query (const QuadratureRole role, const domain::DomainOrSet domain, const MappingKind mapping, const int trialOrder, const int testOrder, const int coefficientOrder, const int geometryWeightOrder, const std::optional< int > baseOrder=std::nullopt) noexcept

8.21.1 Enumeration Type Documentation

8.21.1.1 MappingKind

enum class serif::discretization::quadrature::MappingKind [strong]

Enumerator

none	
affine	
general	
kelvin	

8.21.1.2 Mode

enum class serif::discretization::quadrature::Mode [strong]

Enumerator

fast	
production	
reference	
convergence	

8.21.1.3 QuadratureRole

enum class serif::discretization::quadrature::QuadratureRole [strong]

Enumerator

discretization	
preconditioner	
diagnostic	
projection	

8.21.2 Function Documentation

8.21.2.1 make_query()

```
template<forms::Form FormT>
Query serif::discretization::quadrature::make_query (
                const QuadratureRole role,
                const domain::DomainOrSet domain,
                const MappingKind mapping,
                const int trialOrder,
                const int testOrder,
                const int coefficientOrder,
                const int geometryWeightOrder,
                const std::optional< int > baseOrder = std::nullopt)   [nodiscard], [constexpr], [noexcept]
```

8.22 serif::discretization::quadrature::backend Namespace Reference

Namespaces

- namespace mfem

8.23 serif::discretization::quadrature::backend::mfem Namespace Reference

Classes

- struct DiscreteFunctionalSpaceTraits
- struct DiscreteFunctionalSpaceTraits< functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > >
- struct MFEMOperandListOrder
- struct MFEMOperandListOrder< utils::types::TypeList< OperandTs... > >
- struct MFEMQuadratureContext
- class MFEMResolver
- struct MFEMRule

Concepts

- concept HasReferenceSetIntegrationRule
- concept HasPointerSetIntegrationRule
- concept HasSetIntRule
- concept ConfigurableIntegrator

Typedefs

- template<blocks::HasFunctionalSpace TermT>
  using FunctionalSpaceTraits = DiscreteFunctionalSpaceTraits<typename TermT::FunctionalSpace>

Functions

- template<ConfigurableIntegrator IntegratorT>
  void set_integration_rule (IntegratorT &integrator, const ::mfem::IntegrationRule &integration_rule)
- template<forms::IsOperand OperandT>
  consteval int resolve_operand_order () noexcept
- int sum_dynamic_orders (const std::span< const int > dynamic_orders)

Variables

- template<forms::Form FormT>
  constexpr int mfem_form_static_order = MFEMOperandListOrder<typename FormT::operand_types>::value

8.23.1 Typedef Documentation

8.23.1.1 FunctionalSpaceTraits

template<blocks::HasFunctionalSpace TermT>
using serif::discretization::quadrature::backend::mfem::FunctionalSpaceTraits = DiscreteFunctionalSpaceTraits<typename TermT::FunctionalSpace>

8.23.2 Function Documentation

8.23.2.1 resolve_operand_order()

template<forms::IsOperand OperandT>
int serif::discretization::quadrature::backend::mfem::resolve_operand_order () [nodiscard], [consteval], [noexcept]

8.23.2.2 set_integration_rule()

template<ConfigurableIntegrator IntegratorT>
void serif::discretization::quadrature::backend::mfem::set_integration_rule (
IntegratorT & integrator,
const ::mfem::IntegrationRule & integration_rule)

8.23.2.3 sum_dynamic_orders()

int serif::discretization::quadrature::backend::mfem::sum_dynamic_orders (
const std::span< const int > dynamic_orders) [inline], [nodiscard]

8.23.3 Variable Documentation

8.23.3.1 mfem_form_static_order

template<forms::Form FormT>
int serif::discretization::quadrature::backend::mfem::mfem_form_static_order = MFEMOperandListOrder<typename Form↩
T::operand_types>::value [inline], [constexpr]

8.24 serif::eos Namespace Reference

Namespaces

- namespace models
- namespace relations
- namespace runtime

Classes

- class EOSEvaluationError

Functions

- template<dimensions::ThermodynamicQuantityType OutputQ, models::EOSModel EOS, dimensions::QuantityValue↩ Type... InputValues>
  requires models::EOSSupportingRelation<EOS, relations::Relation<OutputQ, dimensions::QuantityOfT<Input↩ Values>...>>
  constexpr dimensions::QuantityValue< OutputQ > evaluate (const EOS &eos, const InputValues... inputValues) noexcept(noexcept(eos.evaluate(relations::Relation< OutputQ, dimensions::QuantityOfT< InputValues >... >{}, inputValues...)))
- template<dimensions::ThermodynamicQuantityType OutputQ, dimensions::ThermodynamicQuantityType InputQ, models::EOSModel EOS, dimensions::QuantityValueType... InputValues>
  requires models::EOSSupportingPartialDerivative<EOS, relations::Relation<OutputQ, dimensions::QuantityOfT<Input↩ Values>...>, InputQ>
  constexpr dimensions::PartialDerivative< OutputQ, InputQ > partial_derivative (const EOS &eos, const InputValues... inputValues) noexcept(noexcept(eos.partial_derivative(relations::Relation< OutputQ, dimensions::QuantityOfT< InputValues >... >{}, dimensions::WithRespectTo< InputQ >{}, inputValues...)))

8.24.1 Function Documentation

8.24.1.1 evaluate()

template<dimensions::ThermodynamicQuantityType OutputQ, models::EOSModel EOS, dimensions::QuantityValueType... InputValues>
requires models::EOSSupportingRelation<EOS, relations::Relation<OutputQ, dimensions::QuantityOfT<InputValues>...>>
dimensions::QuantityValue< OutputQ > serif::eos::evaluate (
const EOS & eos,
const InputValues... inputValues) [nodiscard], [constexpr], [noexcept]

8.24.1.2 partial_derivative()

template<dimensions::ThermodynamicQuantityType OutputQ, dimensions::ThermodynamicQuantityType InputQ, models::↩ EOSModel EOS, dimensions::QuantityValueType... InputValues>
requires models::EOSSupportingPartialDerivative<EOS, relations::Relation<OutputQ, dimensions::QuantityOfT<Input↩ Values>...>, InputQ>
dimensions::PartialDerivative< OutputQ, InputQ > serif::eos::partial_derivative (
const EOS & eos,
const InputValues... inputValues) [nodiscard], [constexpr], [noexcept]

8.25 serif::eos::models Namespace Reference

Classes

- struct ImplementsPartialDerivative_t
- struct ImplementsRelation_t
- struct ImplementsRelationCatalog_t
- struct ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > >
- struct IsEquationOfStateModel_t
- struct IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std::remove_cvref_t< EOSCandidate >::
- class Polytrope
- struct PolytropeParameters

Concepts

- concept EOSModel
- concept EOSSupportingRelation
- concept EOSSupportingPartialDerivative
- concept StructureSeedEOS
- concept BarotropicClosureEOS
- concept PressureForceEOS

Enumerations

- enum class SERiFErrorCode

Functions

- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsNumeric Scalar>
  void validate_finite (const Scalar value)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void validate_greater_than (const Scalar value, const Scalar predicate)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void validate_less_than (const Scalar value, const Scalar predicate)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void validate_nonnegativity (const Scalar value)

Variables

- template<typename EOSCandidate>
  constexpr bool is_equation_of_state_model_v = IsEquationOfStateModel_t<std::remove_cvref_t<EOSCandidate>>::value
- template<typename EOS, typename RelationT>
  constexpr bool supports_relation_v = is_equation_of_state_model_v<EOS> && ImplementsRelation_t<std::remove_cvref_t<EOS>, std::remove_cvref_t<RelationT>>::value
- template<typename EOS, typename RelationT, typename InputQuantity>
  constexpr bool supports_partial_derivative_v = is_equation_of_state_model_v<EOS> && ImplementsPartialDerivative_t<std::remove_cvref_t<EOS>, std::remove_cvref_t<RelationT>, InputQuantity>::value

8.25.1 Enumeration Type Documentation

8.25.1.1 SERiFErrorCode

enum class serif::utils::errors::SERiFErrorCode : uint16_t [strong]

8.25.2 Function Documentation

8.25.2.1 validate_finite()

```
template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsNumeric Scalar>
void serif::utils::misc::validate_finite (
                const Scalar value)
```

8.25.2.2 validate_greater_than()

```
template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
void serif::utils::misc::validate_greater_than (
                const Scalar value,
                const Scalar predicate)
```

8.25.2.3 validate_less_than()

```
template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
void serif::utils::misc::validate_less_than (
                const Scalar value,
                const Scalar predicate)
```

8.25.2.4 validate_nonnegativity()

```
template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
void serif::utils::misc::validate_nonnegativity (
                const Scalar value)
```

8.25.3 Variable Documentation

8.25.3.1 is_equation_of_state_model_v

```
template<typename EOSCandidate>
bool serif::eos::models::is_equation_of_state_model_v = IsEquationOfStateModel_t<std::remove_cvref_t<EOSCandidate>>↩
::value  [inline], [constexpr]
```

8.25.3.2 supports_partial_derivative_v

```
template<typename EOS, typename RelationT, typename InputQuantity>
bool serif::eos::models::supports_partial_derivative_v = is_equation_of_state_model_v<EOS> && ImplementsPartialDerivative_t<std↩
::remove_cvref_t<EOS>, std::remove_cvref_t<RelationT>, InputQuantity>::value  [inline], [constexpr]
```

8.25.3.3 supports_relation_v

template<typename EOS, typename RelationT>
bool serif::eos::models::supports_relation_v = is_equation_of_state_model_v<EOS> && ImplementsRelation_t<std::remove_cvref_t<EOS>, std::remove_cvref_t<RelationT>>::value [inline], [constexpr]

8.26 serif::eos::relations Namespace Reference

Classes

- struct CatalogContainsRelation
- struct CatalogContainsRelation< RelationCatalog< Relations... >, RelationT >
- struct IsThermodynamicRelation
- struct IsThermodynamicRelation< Relation< Output, Inputs... > >
- struct IsValidRelationCatalog
- struct IsValidRelationCatalog< RelationCatalog< Relations... > >
- struct Relation
- struct RelationCatalog
- struct RelationContainsInput
- struct RelationContainsInput< Relation< Output, Inputs... >, Quantity >

Concepts

- concept CatalogContainingRelation
- concept RelationContainingInput
- concept ValidThermodynamicRelation
- concept ValidRelationCatalog

Typedefs

- using PressureFromDensity
- using PressureFromSpecificEnthalpy
- using SpecificEnthalpyFromDensity
- using SpecificEnthalpyFromPressure
- using DensityFromSpecificEnthalpy

Functions

- template<typename... Ts>
  consteval std::size_t CountVariadicArguments () noexcept

Variables

- template<typename Catalog, typename RelationT>
  constexpr bool relation_catalog_contains_relation_v = CatalogContainsRelation<std::remove_cv_t<Catalog>, std::remove_cv_t<RelationT>>::value
- template<typename RelationT, typename Quantity>
  constexpr bool relation_contains_input_v = RelationContainsInput<std::remove_cv_t<RelationT>, std::remove_cv_t<Quantity>>::value
- template<typename RelationCandidate>
  constexpr bool is_thermodynamic_relation_v = IsThermodynamicRelation<RelationCandidate>::value
- template<typename CatalogCandidate>
  constexpr bool is_valid_relation_catalog_v = IsValidRelationCatalog<CatalogCandidate>::value

8.26.1 Typedef Documentation

8.26.1.1 DensityFromSpecificEnthalpy

using serif::eos::relations::DensityFromSpecificEnthalpy

Initial value:

```
Relation<
    dimensions::MassDensity,
    dimensions::SpecificEnthalpy>
```

8.26.1.2 PressureFromDensity

using serif::eos::relations::PressureFromDensity

Initial value:

```
Relation<
    dimensions::Pressure,
    dimensions::MassDensity>
```

8.26.1.3 PressureFromSpecificEnthalpy

using serif::eos::relations::PressureFromSpecificEnthalpy

Initial value:

```
Relation<
    dimensions::Pressure,
    dimensions::SpecificEnthalpy>
```

8.26.1.4 SpecificEnthalpyFromDensity

using serif::eos::relations::SpecificEnthalpyFromDensity

Initial value:

```
Relation<
    dimensions::SpecificEnthalpy,
    dimensions::MassDensity>
```

8.26.1.5 SpecificEnthalpyFromPressure

using serif::eos::relations::SpecificEnthalpyFromPressure

Initial value:

```
Relation<
    dimensions::SpecificEnthalpy,
    dimensions::Pressure>
```

8.26.2 Function Documentation

8.26.2.1 CountVariadicArguments()

template<typename... Ts>
std::size_t serif::utils::misc::std_helper::CountVariadicArguments () [nodiscard], [consteval], [noexcept]

8.26.3 Variable Documentation

8.26.3.1 is_thermodynamic_relation_v

template<typename RelationCandidate>
bool serif::eos::relations::is_thermodynamic_relation_v = IsThermodynamicRelation<RelationCandidate>::value [constexpr]

8.26.3.2 is_valid_relation_catalog_v

template<typename CatalogCandidate>
bool serif::eos::relations::is_valid_relation_catalog_v = IsValidRelationCatalog<CatalogCandidate>::value [constexpr]

8.26.3.3 relation_catalog_contains_relation_v

template<typename Catalog, typename RelationT>
bool serif::eos::relations::relation_catalog_contains_relation_v = CatalogContainsRelation<std::remove_cv_t<Catalog>, std::remove_cv_t<RelationT>>::value [inline], [constexpr]

8.26.3.4 relation_contains_input_v

template<typename RelationT, typename Quantity>
bool serif::eos::relations::relation_contains_input_v = RelationContainsInput<std::remove_cv_t<RelationT>, std::remove_cv_t<Quantity>>::value [inline], [constexpr]

8.27 serif::eos::runtime Namespace Reference

Classes

- class EOSView
- struct HasRuntimeQuantityIdentifiers
- struct HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > >
- struct RuntimeCatalogDispatch
- struct RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > >
- struct RuntimeCatalogIsSupported
- struct RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > >
- struct RuntimeCatalogStorage
- struct RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > >
- struct RuntimeRelationQuantities
- struct RuntimeRelationQuantities< relations::Relation< OutputT, InputTs... > >
- struct RuntimeRelationsAreSupported
- struct RuntimeRelationsAreSupported< true, RelationTs... >
- struct RuntimeRelationStorage
- struct RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >

Concepts

- concept RuntimeEOSModel

Typedefs

- template<RuntimeEOSModel EOSCandidate>
  using RuntimeEOSAdapter = RuntimeCatalogDispatch<EOSCandidate, typename EOSCandidate::↩
  Relations>
- template<typename... RelationTs>
  using RuntimeCatalogQuantityTuple = decltype(std::tuple_cat(std::declval<typename RuntimeRelationQuantities<Re
  Ts>::Type>()...))

Functions

- template<RuntimeEOSModel EOSCandidate>
  constexpr std::span< const dimensions::runtime::RuntimeRelationDescriptor > runtimeRelationDescriptors
  () noexcept
- template<typename FirstQuantityT, typename SecondQuantityT>
  consteval bool runtimeQuantityIdentifiersAreCompatible ()
- template<typename QuantityTupleT, std::size_t First, std::size_t... Offsets>
  consteval bool runtimeQuantityIdentifierIsUnambiguous (std::index_sequence< Offsets... >)
- template<typename QuantityTupleT, std::size_t... Indices>
  consteval bool runtimeQuantityIdentifiersAreUnambiguous (std::index_sequence< Indices... >)
- std::expected< double, utils::errors::SERiFErrorCode > runtimeEvaluationFailure (const utils::errors::SERiFErrorCode
  code)

- template<typename EOS, typename OutputT, typename... InputTs>
  std::expected< double, utils::errors::SERiFErrorCode > evaluateRuntimeRelation (const EOS &eos, relations::Relation< OutputT, InputTs... >, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues)
- template<typename InputQuantityT, typename EOS, typename OutputT, typename... InputTs>
  bool tryRuntimePartialDerivative (const EOS &eos, relations::Relation< OutputT, InputTs... >, const dimensions::runtime::ThermodynamicQuantityID withRespectTo, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues, std::expected< double, utils::errors::SERiFErrorCode > &result)
- template<typename EOS, typename OutputT, typename... InputTs>
  std::expected< double, utils::errors::SERiFErrorCode > evaluateRuntimePartialDerivative (const EOS &eos, relations::Relation< OutputT, InputTs... > relation, const dimensions::runtime::ThermodynamicQuantityID withRespectTo, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues)
- template<typename EOS, typename RelationT>
  bool runtimeRelationMatches (const dimensions::runtime::ThermodynamicQuantityID outputQuantityID, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues)

8.27.1 Typedef Documentation

8.27.1.1 RuntimeCatalogQuantityTuple

```
template<typename... RelationTs>
using serif::eos::runtime::RuntimeCatalogQuantityTuple = decltype(std::tuple_cat(std::declval<typename RuntimeRelationQuantities<RelationTs>::Type>()...))
```

8.27.1.2 RuntimeEOSAdapter

```
template<RuntimeEOSModel EOSCandidate>
using serif::eos::runtime::RuntimeEOSAdapter = RuntimeCatalogDispatch<EOSCandidate, typename EOSCandidate::Relations>
```

8.27.2 Function Documentation

8.27.2.1 evaluateRuntimePartialDerivative()

```
template<typename EOS, typename OutputT, typename... InputTs>
std::expected< double, utils::errors::SERiFErrorCode > serif::eos::runtime::evaluateRuntimePartialDerivative (
        const EOS & eos,
        relations::Relation< OutputT, InputTs... > relation,
        const dimensions::runtime::ThermodynamicQuantityID withRespectTo,
        const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues)   [nodiscard]
```

8.27.2.2 evaluateRuntimeRelation()

```
template<typename EOS, typename OutputT, typename... InputTs>
std::expected< double, utils::errors::SERiFErrorCode > serif::eos::runtime::evaluateRuntimeRelation (
        const EOS & eos,
        relations::Relation< OutputT, InputTs... > ,
        const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues)   [nodiscard]
```

8.27.2.3 runtimeEvaluationFailure()

std::expected< double, utils::errors::SERiFErrorCode > serif::eos::runtime::runtimeEvaluationFailure (
const utils::errors::SERiFErrorCode code) [inline], [nodiscard]

8.27.2.4 runtimeQuantityIdentifierIsUnambiguous()

template<typename QuantityTupleT, std::size_t First, std::size_t... Offsets>
bool serif::eos::runtime::runtimeQuantityIdentifierIsUnambiguous (
std::index_sequence< Offsets... >) [nodiscard], [consteval]

8.27.2.5 runtimeQuantityIdentifiersAreCompatible()

template<typename FirstQuantityT, typename SecondQuantityT>
bool serif::eos::runtime::runtimeQuantityIdentifiersAreCompatible () [nodiscard], [consteval]

8.27.2.6 runtimeQuantityIdentifiersAreUnambiguous()

template<typename QuantityTupleT, std::size_t... Indices>
bool serif::eos::runtime::runtimeQuantityIdentifiersAreUnambiguous (
std::index_sequence< Indices... >) [nodiscard], [consteval]

8.27.2.7 runtimeRelationDescriptors()

template<RuntimeEOSModel EOSCandidate>
std::span< const dimensions::runtime::RuntimeRelationDescriptor > serif::eos::runtime::runtimeRelationDescriptors () [nodiscard], [constexpr], [noexcept]

8.27.2.8 runtimeRelationMatches()

template<typename EOS, typename RelationT>
bool serif::eos::runtime::runtimeRelationMatches (
const dimensions::runtime::ThermodynamicQuantityID outputQuantityID,
const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues) [nodiscard]

8.27.2.9 tryRuntimePartialDerivative()

template<typename InputQuantityT, typename EOS, typename OutputT, typename... InputTs>
bool serif::eos::runtime::tryRuntimePartialDerivative (
const EOS & eos,
relations::Relation< OutputT, InputTs... > ,
const dimensions::runtime::ThermodynamicQuantityID withRespectTo,
const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues,
std::expected< double, utils::errors::SERiFErrorCode > & result) [nodiscard]

8.28 serif::equations Namespace Reference

Namespaces

- namespace gravity_contributions

Classes

- class AngularMomentum

  Close the unknown angular speed with integral_star rho Omega s^2 dV - target_momentum = 0.
- class BarotropicClosure
- class BarotropicHydrostaticEquilibrium
- struct CenteringConstraint
- class CentralDensityConstraint

  Prescribe central density through the continuous enthalpy: h(0) - h_EOS(rho_c) = 0.
- struct Contribution

  Optional authoring base: a test operation and the domains of one weak contribution.
- struct Contributions

  Collect leaf equation types without copying the objects in a composition.
- struct Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions())> >
- struct ContributionSupport
- struct ContributionSupport< T, std::void_t< typename T::Support > >
- struct ContributionTuple
- struct ContributionTuple< std::tuple< Ts... >, std::void_t< decltype(std::tuple_cat(std::declval< typename Contribu
- class DisplacementForce

  Mechanical balance: pressure force + gravity force + rotational force.
- class EquationDifference

  Subtract every weak contribution of the right equation from the left equation.
- struct EquationPoint
- class EquationRegistry
- class EquationScale

  Multiply an equation's weak contributions by a constant independent of state and position.
- class EquationSum

  Add equations sharing a residual row; each retains its own test operation and support.
- struct GravityForce
- struct GravityGradientRelation
- class MassNormalization

  Close the unknown Bernoulli constant with integral_star rho dV - target_mass = 0.
- struct OperandValues
- struct OperandValues< utils::types::TypeList< Operands... >, OutputT >
- struct PointLinearization
- class PoissonBalance
- class PressureForce

  Pressure's contribution to displacement balance: -integral_star P(h) div(w) dV.
- class PressureSurfaceConstraint

  Prescribed pressure on the stellar boundary: h - h_EOS(P_surface) = 0.
- class RigidRotation
- class RotationalForce

  Rotational contribution to displacement balance: -integral_star rho grad(Psi) . w dV.
- struct TestOperandTraits
- struct TestOperandTraits< T, std::void_t< typename T::TestOperand > >
- struct ValidContributions
- struct ValidContributions< RowT, std::tuple< C... > >

Concepts

- concept PointEquation
- concept Equation

  Physics authoring contract; contributions share one residual row but may use different tests/supports.

Typedefs

- template<typename T>
  using TestOperandOf = typename TestOperandTraits<T>::type
- template<typename T>
  using SupportOf = typename ContributionSupport<T>::type
- template<typename T>
  using ContributionsOf = typename Contributions<std::remove_cvref_t<T>>::type

8.28.1 Typedef Documentation

8.28.1.1 ContributionsOf

template<typename T>
using serif::equations::ContributionsOf = typename Contributions<std::remove_cvref_t<T>>::type

8.28.1.2 SupportOf

template<typename T>
using serif::equations::SupportOf = typename ContributionSupport<T>::type

8.28.1.3 TestOperandOf

template<typename T>
using serif::equations::TestOperandOf = typename TestOperandTraits<T>::type

8.29 serif::equations::gravity_contributions Namespace Reference

Classes

- struct DensitySource
- struct GradientDivergence
- struct GradientValue
- struct PotentialValue

Typedefs

- using Gradient = discretization::blocks::gravity::gradient
- using Potential = discretization::blocks::gravity::potential
- using Density = discretization::blocks::density::mass
- using Value = discretization::forms::Operation::Value
- using Divergence = discretization::forms::Operation::Divergence
- template<typename Term, typename Operation = Value>
  using Operand = discretization::forms::Operand<Term, Operation>

8.29.1 Typedef Documentation

8.29.1.1 Density

using serif::equations::gravity_contributions::Density = discretization::blocks::density::mass

8.29.1.2 Divergence

using serif::equations::gravity_contributions::Divergence = discretization::forms::Operation::Divergence

8.29.1.3 Gradient

using serif::equations::gravity_contributions::Gradient = discretization::blocks::gravity::gradient

8.29.1.4 Operand

template<typename Term, typename Operation = Value>
using serif::equations::gravity_contributions::Operand = discretization::forms::Operand<Term, Operation>

8.29.1.5 Potential

using serif::equations::gravity_contributions::Potential = discretization::blocks::gravity::potential

8.29.1.6 Value

using serif::equations::gravity_contributions::Value = discretization::forms::Operation::Value

8.30 serif::geometry Namespace Reference

Namespaces

- namespace deformation
- namespace surface

8.31 serif::geometry::deformation Namespace Reference

Namespaces

- namespace backend
- namespace detail
- namespace keys
- namespace spaces

Classes

- class ComposedDeformation

  Compose shape-to-surface and surface-to-volume maps with the complete derivative chain rule.
- struct DeformationFailure

  Describe a rejected geometry calculation; the dependency cache publishes results only after success.
- struct DeformationLinearization

  Freeze the shape coordinates associated with a derivative, independently of mutable caller inputs.
- struct DeformationWorkspaceType

  Select numerical workspace storage without adding a preparation owner.
- struct DeformationWorkspaceType< Map, true >
- struct DomainDeformation

  State how the stellar surface moves and how that motion extends into the two surrounding regions.
- struct EmptyDeformationWorkspace

  Stateless linear actions need no hidden mutable scratch.
- struct FixedInfinityRadialExtension

  Reduce surface motion linearly to zero at reference infinity in logical reference radius.
- struct GeometryStepLimit

  Geometry-only bound at the supplied sample points, with a location for the first obstruction.
- struct GeometryStepOptions

  Numerical controls for keeping a straight volume-displacement step above a Jacobian determinant floor.
- struct NodalRadialSurface

  Give each surface node an independent radial displacement, measured in physical length.
- struct PowerLawRadialExtension

  Extend surface motion toward a fixed stellar center using a power of logical reference radius.
- class RadialExtension

  Extend one interleaved surface vector into the MFEM component-block volume layout. Stellar ownership wins at shared nodes. Both actions use the same scalar interpolation, so their pairing is Euclidean virtual work without an additional surface-area weight.
- struct RadialInterpolation

  Own the surface interpolation shared by the stellar and vacuum extensions.
- struct ReferenceSurface

  Own fixed surface positions and their canonical ascending scalar-node indices.
- struct SphericalHarmonicMode

  Identify a real harmonic, with negative orders using sine and positive orders using cosine.
- struct SphericalHarmonicSurface

  Represent radial displacement lengths with normalized real spherical harmonics on the reference surface.
- class SurfaceEmbedding

  Own a linear conversion from radial displacement lengths to Cartesian surface displacements.

Concepts

- concept LinearDeformationMap

  Fixed maps supply their forward and work-conjugate transpose once.
- concept DifferentiableDeformationMap

  Nonlinear maps explicitly provide every term needed by a Newton force pullback.
- concept DeformationMap

Typedefs

- template<DeformationMap Map>
  using DeformationWorkspace = typename DeformationWorkspaceType<Map>::Type
- using DeformationStatus = std::expected<void, DeformationFailure>

Enumerations

- enum class DeformationFailureReason {
  invalid_configuration , incompatible_discretization , invalid_size , non_finite ,
  unresolved_surface , invalid_geometry }

Functions

- DeformationStatus validate_deformation_values (std::span< const double > values, std::size_t expected_size)

  Reject invalid mathematical arguments before a map touches its output.
- template<DeformationMap Map>
  DeformationWorkspace< Map > make_deformation_workspace (const Map &map)

  Allocate action scratch explicitly, once per independent caller.
- template<DeformationMap Map>
  DeformationStatus build_displacement (const Map &map, std::span< const double > parameters, std::span< double > displacement, DeformationWorkspace< Map > &workspace)

  Evaluate physical displacement with caller-owned output and scratch. All action outputs must be disjoint from their inputs. Workspace storage is reserved for the map and must not supply public input or output spans.
- template<DeformationMap Map>
  DeformationStatus apply_jacobian (const Map &map, std::span< const double > parameters, std::span< const double > direction, std::span< double > displacement_direction, DeformationWorkspace< Map > &workspace)

  Linear maps reuse their primal action; nonlinear maps supply their exact tangent.
- template<DeformationMap Map>
  DeformationStatus apply_jacobian_transpose (const Map &map, std::span< const double > parameters, std::span< const double > displacement_dual, std::span< double > parameter_dual, DeformationWorkspace< Map > &workspace)

  Transfer virtual work from the displacement field to its independent shape coefficients.
- template<DeformationMap Map>
  DeformationStatus apply_pullback_derivative (const Map &map, std::span< const double > parameters, std::span< const double > direction, std::span< const double > displacement_dual, std::span< double > parameter_dual, DeformationWorkspace< Map > &workspace)

  Differentiate the transpose at fixed dual; zero is justified only for a declared linear map.

- template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
  requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
  std::expected< ComposedDeformation< SurfaceMap, ExtensionMap >, DeformationFailure > compose_deformation (std::shared_ptr< const SurfaceMap > surface, std::shared_ptr< const ExtensionMap > extension)

  Validate factor sizes before constructing an owning mathematical composition.
- template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
  requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
  auto make_deformation_computations ()

  Declare the numerical composition as ordinary dependency computations.
- template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
  requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
  auto make_deformation_graph ()

  Compose independently compiled mathematical factors without introducing a prepared-operator facade.
- template<DeformationMap Map>
  std::expected< std::vector< double >, DeformationFailure > evaluate_displacement (const Map &map, std::span< const double > parameters)

  Allocate a result for graph publication while leaving repeated low-level actions allocation-free.
- template<DeformationMap Map>
  DeformationStatus apply_jacobian (const DeformationLinearization< Map > &state, std::span< const double > direction, std::span< double > output, DeformationWorkspace< Map > &workspace)

  Evaluate a tangent about the cache's accepted shape, not the caller's next trial state.
- template<DeformationMap Map>
  DeformationStatus apply_jacobian_transpose (const DeformationLinearization< Map > &state, std::span< const double > dual, std::span< double > output, DeformationWorkspace< Map > &workspace)

  Pull displacement forces back to the accepted independent shape coordinates.
- template<DeformationMap Map>
  DeformationStatus apply_pullback_derivative (const DeformationLinearization< Map > &state, std::span< const double > direction, std::span< const double > dual, std::span< double > output, DeformationWorkspace< Map > &workspace)

  Keep the geometric contribution to the Newton force derivative available for nonlinear models.
- std::expected< RadialExtension, DeformationFailure > make_radial_extension (std::shared_ptr< const RadialInterpolation > interpolation, PowerLawRadialExtension interior={}, FixedInfinityRadialExtension vacuum={})

  Compile regional weights while sharing the immutable surface interpolation. Callers supplying a shared pointer must treat its data as immutable after compilation; the dependency graph publishes a const owning value for precisely this reason.
- std::expected< SurfaceEmbedding, DeformationFailure > build_surface_embedding (const NodalRadialSurface &model, std::span< const std::array< double, 3 > > positions)

  Compile fixed radial directions from ordered physical reference positions without retaining backend objects.
- std::expected< SurfaceEmbedding, DeformationFailure > build_surface_embedding (const SphericalHarmonicSurface &model, std::span< const std::array< double, 3 > > positions)

  Sample physically normalized real harmonics and reject unresolved modes without renormalizing or dropping coordinates.

Variables

- template<DeformationMap Map>
  constexpr bool is_linear_deformation

  Linearity is a property of the numerical map, independent of the cache lifecycle.

8.31.1 Typedef Documentation

8.31.1.1 DeformationStatus

using serif::geometry::deformation::DeformationStatus = std::expected<void, DeformationFailure>

8.31.1.2 DeformationWorkspace

template<DeformationMap Map>
using serif::geometry::deformation::DeformationWorkspace = typename DeformationWorkspaceType<Map>::Type

8.31.2 Enumeration Type Documentation

8.31.2.1 DeformationFailureReason

enum class serif::geometry::deformation::DeformationFailureReason [strong]

Enumerator

invalid_configuration	
incompatible_discretization	
invalid_size	
non_finite	
unresolved_surface	
invalid_geometry	

8.31.3 Function Documentation

8.31.3.1 apply_jacobian() [1/2]

template<DeformationMap Map>
DeformationStatus serif::geometry::deformation::apply_jacobian (
const DeformationLinearization< Map > & state,
std::span< const double > direction,
std::span< double > output,
DeformationWorkspace< Map > & workspace) [nodiscard]

Evaluate a tangent about the cache's accepted shape, not the caller's next trial state.

8.31.3.2 apply_jacobian() [2/2]

```
template<DeformationMap Map>
DeformationStatus serif::geometry::deformation::apply_jacobian (
        const Map & map,
        std::span< const double > parameters,
        std::span< const double > direction,
        std::span< double > displacement_direction,
        DeformationWorkspace< Map > & workspace)   [nodiscard]
```

Linear maps reuse their primal action; nonlinear maps supply their exact tangent.

8.31.3.3 apply_jacobian_transpose() [1/2]

```
template<DeformationMap Map>
DeformationStatus serif::geometry::deformation::apply_jacobian_transpose (
        const DeformationLinearization< Map > & state,
        std::span< const double > dual,
        std::span< double > output,
        DeformationWorkspace< Map > & workspace)   [nodiscard]
```

Pull displacement forces back to the accepted independent shape coordinates.

8.31.3.4 apply_jacobian_transpose() [2/2]

```
template<DeformationMap Map>
DeformationStatus serif::geometry::deformation::apply_jacobian_transpose (
        const Map & map,
        std::span< const double > parameters,
        std::span< const double > displacement_dual,
        std::span< double > parameter_dual,
        DeformationWorkspace< Map > & workspace)   [nodiscard]
```

Transfer virtual work from the displacement field to its independent shape coefficients.

8.31.3.5 apply_pullback_derivative() [1/2]

```
template<DeformationMap Map>
DeformationStatus serif::geometry::deformation::apply_pullback_derivative (
        const DeformationLinearization< Map > & state,
        std::span< const double > direction,
        std::span< const double > dual,
        std::span< double > output,
        DeformationWorkspace< Map > & workspace)   [nodiscard]
```

Keep the geometric contribution to the Newton force derivative available for nonlinear models.

8.31.3.6 apply_pullback_derivative() [2/2]

```
template<DeformationMap Map>
DeformationStatus serif::geometry::deformation::apply_pullback_derivative (
        const Map & map,
        std::span< const double > parameters,
        std::span< const double > direction,
        std::span< const double > displacement_dual,
        std::span< double > parameter_dual,
        DeformationWorkspace< Map > & workspace)   [nodiscard]
```

Differentiate the transpose at fixed dual; zero is justified only for a declared linear map.

8.31.3.7 build_displacement()

```
template<DeformationMap Map>
DeformationStatus serif::geometry::deformation::build_displacement (
        const Map & map,
        std::span< const double > parameters,
        std::span< double > displacement,
        DeformationWorkspace< Map > & workspace)   [nodiscard]
```

Evaluate physical displacement with caller-owned output and scratch. All action outputs must be disjoint from their inputs. Workspace storage is reserved for the map and must not supply public input or output spans.

8.31.3.8 build_surface_embedding() [1/2]

```
std::expected< SurfaceEmbedding, DeformationFailure > serif::geometry::deformation::build_surface_embedding (
        const NodalRadialSurface & model,
        std::span< const std::array< double, 3 > > positions)   [nodiscard]
```

Compile fixed radial directions from ordered physical reference positions without retaining backend objects.

8.31.3.9 build_surface_embedding() [2/2]

```
std::expected< SurfaceEmbedding, DeformationFailure > serif::geometry::deformation::build_surface_embedding (
        const SphericalHarmonicSurface & model,
        std::span< const std::array< double, 3 > > positions)   [nodiscard]
```

Sample physically normalized real harmonics and reject unresolved modes without renormalizing or dropping coordinates.

8.31.3.10 compose_deformation()

```
template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
std::expected< ComposedDeformation< SurfaceMap, ExtensionMap >, DeformationFailure > serif::geometry::deformation↩
::compose_deformation (
                std::shared_ptr< const SurfaceMap > surface,
                std::shared_ptr< const ExtensionMap > extension)   [nodiscard]
```

Validate factor sizes before constructing an owning mathematical composition.

8.31.3.11 evaluate_displacement()

```
template<DeformationMap Map>
std::expected< std::vector< double >, DeformationFailure > serif::geometry::deformation::evaluate_displacement (
                const Map & map,
                std::span< const double > parameters)   [nodiscard]
```

Allocate a result for graph publication while leaving repeated low-level actions allocation-free.

8.31.3.12 make_deformation_computations()

```
template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
auto serif::geometry::deformation::make_deformation_computations ()   [nodiscard]
```

Declare the numerical composition as ordinary dependency computations.

Note

Shared immutable factors, coefficient snapshots, and results belong to the dependency cache. Analytical actions remain independent functions.

8.31.3.13 make_deformation_graph()

```
template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
auto serif::geometry::deformation::make_deformation_graph ()   [nodiscard]
```

Compose independently compiled mathematical factors without introducing a prepared-operator facade.

8.31.3.14 make_deformation_workspace()

```
template<DeformationMap Map>
DeformationWorkspace< Map > serif::geometry::deformation::make_deformation_workspace (
                const Map & map)   [nodiscard]
```

Allocate action scratch explicitly, once per independent caller.

8.31.3.15 make_radial_extension()

```
std::expected< RadialExtension, DeformationFailure > serif::geometry::deformation::make_radial_extension (
                std::shared_ptr< const RadialInterpolation > interpolation,
                PowerLawRadialExtension interior = {},
                FixedInfinityRadialExtension vacuum = {})   [nodiscard]
```

Compile regional weights while sharing the immutable surface interpolation. Callers supplying a shared pointer must treat its data as immutable after compilation; the dependency graph publishes a const owning value for precisely this reason.

8.31.3.16 validate_deformation_values()

```
DeformationStatus serif::geometry::deformation::validate_deformation_values (
                std::span< const double > values,
                std::size_t expected_size)   [inline], [nodiscard]
```

Reject invalid mathematical arguments before a map touches its output.

8.31.4 Variable Documentation

8.31.4.1 is_linear_deformation

```
template<DeformationMap Map>
bool serif::geometry::deformation::is_linear_deformation   [inline], [constexpr]
```

Initial value:

```
= [] {
      if constexpr (requires { { Map::is_linear } -> std::convertible_to<bool>; }) {
          return Map::is_linear;
      } else {
          return false;
      }
  }()
```

Linearity is a property of the numerical map, independent of the cache lifecycle.

8.32 serif::geometry::deformation::backend Namespace Reference

Namespaces

- namespace mfem

8.33 serif::geometry::deformation::backend::mfem Namespace Reference

Namespaces

- namespace detail
- namespace volume_displacement_keys

Classes

- struct DeformationValueTracking

  Model options and coefficient copies describe values independently of wrapper addresses.
- struct GeometrySampleRule

  Borrow one sampling rule; repeat an element to retain the actual union of downstream rules.
- struct ReferenceDiscretization

  Borrow the paired reference meshes; attributes retain their physical domain meaning.
- struct ReferenceDiscretizationSnapshot

  One comparable snapshot covers all inputs used to compile fixed deformation data.
- struct ReferenceDiscretizationTracking

  Let the generic dependency engine track MFEM metadata without owning meshes.
- struct ReferenceMeshSnapshot

  Capture coordinate revisions as well as topology and mutable material/boundary labels.
- struct ReferenceSpaceSnapshot

  Track the coefficient interpretation independently of numerical coefficient values.

Functions

- auto make_surface_metric_computation ()

  Optionally supply the reference surface mass diagonal in native shape coordinates.
- template<typename Surface>
  requires requires(const Surface& model, std::span<const std::array<double, 3>> positions) { { build_surface_embedding(model, positions) } -> std::same_as<std::expected<SurfaceEmbedding, DeformationFailure>>; }
  auto make_deformation_computations ()

  Compile fixed reference factors and evaluate shape motion with one dependency engine.
- template<typename Surface>
  auto make_deformation_graph ()

  Provide a standalone graph while leaving the same computations available for equation composition.
- std::expected< GeometryStepLimit, DeformationFailure > estimate_geometry_step_limit (const mapping::DomainMapper &mapper, const ::mfem::FiniteElementSpace &displacement_space, const ::mfem::GridFunction &compactification_coordinate, const ::mfem::Vector &accepted_displacement, const ::mfem::Vector &displacement_direction, std::span< const GeometrySampleRule > sample_↩ rules, GeometryStepOptions options={})

  Find the first sampled mapping boundary along accepted + step direction.
- template<discretization::domain::schema::IsSchema Schema = discretization::domain::schema::CoreEnvelopeVacuum↩ DomainSchema>
  requires (Schema::template contains_domain<discretization::domain::StellarDomains>() && Schema::template contains_domain<discretization::domain::VacuumDomain>() && Schema::template contains_boundary<discretization::domain::StellarSurf && Schema::template contains_boundary<discretization::domain::InfinitySurfaceBoundary>())
  ReferenceDiscretization deformation_discretization (const ::mfem::FiniteElementSpace &scalar_space, const ::mfem::FiniteElementSpace &displacement_space, ::mfem::Mesh &logical_mesh)

  Resolve physical domain names through the schema rather than hard-coded MFEM attributes.
- template<typename Surface, typename Interior, typename Vacuum>
  auto make_deformation_inputs (const DomainDeformation< Surface, Interior, Vacuum > &model, const ReferenceDiscretization &discretization, const std::vector< double > &coefficients)

  Bind physical choices and shape coordinates without evaluating deformation.
- template<typename Surface, typename Interior, typename Vacuum>
  void make_deformation_inputs (const DomainDeformation< Surface, Interior, Vacuum > &, const ReferenceDiscretization &, const std::vector< double > &&)=delete

Reject temporary coefficient vectors whose storage would expire before preparation.

- std::expected< RadialInterpolation, DeformationFailure > build_radial_interpolation (const ::mfem↩ ::FiniteElementSpace &scalar_space, const ::mfem::FiniteElementSpace &vector_space, ::mfem::Mesh &logical_mesh, std::span< const int > stellar_element_support, std::span< const int > vacuum_↩ element_support, int surface_attribute, int infinity_attribute)

  Compile logical-ray interpolation on paired physical and logical meshes. Supports are zero/one element masks. The logical radius is the maximum absolute Cartesian coordinate, preserving the STROID reference foliation after physical distortion. Only conforming serial nodal spaces and component-block vector ordering are supported.
- DeformationStatus validate_scalar_reference_space (const ::mfem::FiniteElementSpace &scalar_↩ space)

  Reject spaces whose coefficients cannot be interpreted as serial nodal positions.
- DeformationStatus validate_companion_numbering (const ::mfem::FiniteElementSpace &reference_↩ space, const ::mfem::FiniteElementSpace &companion_space)

  Require identical scalar coefficient meanings in two companion spaces. Vector component ordering may differ; element bases and their scalar-node indices must agree.
- std::expected< std::vector< std::array< double, 3 > >, DeformationFailure > sample_reference_positions (const ::mfem::FiniteElementSpace &scalar_space, ::mfem::Mesh &reference_mesh)

  Sample a paired mesh at the scalar companion nodes, checking shared-node agreement.
- std::expected< std::vector< int >, DeformationFailure > select_surface_scalar_dofs (const ::mfem::↩ FiniteElementSpace &scalar_space, int surface_attribute)

  Select canonical surface nodes without retaining a mesh or finite-element-space pointer.
- std::expected< ReferenceSurface, DeformationFailure > sample_reference_surface (const ::mfem::↩ FiniteElementSpace &scalar_space, int surface_attribute)

  Sample physical reference positions in ascending scalar-DOF order.
- std::expected< std::vector< double >, DeformationFailure > surface_parameter_mass_diagonal (const ::mfem::FiniteElementSpace &scalar_space, int surface_attribute, const ReferenceSurface &reference_surface, const SurfaceEmbedding &embedding)

  Measure each shape parameter with the reference boundary mass norm of its Cartesian finite-element trace.
- template<discretization::domain::schema::IsSchema Schema>
  auto make_volume_displacement_computations (const discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, Schema > &binding)

  Connect a full component-block deformation vector to an equation's displacement binding.
- template<discretization::domain::schema::IsSchema Schema>
  void make_volume_displacement_computations (const discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, Schema > &&)=delete

  A graph cannot retain a displacement binding supplied as a temporary.

8.33.1 Function Documentation

8.33.1.1 build_radial_interpolation()

std::expected< RadialInterpolation, DeformationFailure > serif::geometry::deformation::backend::mfem::build_radial_↩ interpolation (

const ::mfem::FiniteElementSpace & scalar_space,
const ::mfem::FiniteElementSpace & vector_space,
::mfem::Mesh & logical_mesh,
std::span< const int > stellar_element_support,
std::span< const int > vacuum_element_support,

int surface_attribute,
int infinity_attribute) [nodiscard]

Compile logical-ray interpolation on paired physical and logical meshes. Supports are zero/one element masks. The logical radius is the maximum absolute Cartesian coordinate, preserving the STROID reference foliation after physical distortion. Only conforming serial nodal spaces and component-block vector ordering are supported.

8.33.1.2 deformation_discretization()

template<discretization::domain::schema::IsSchema Schema = discretization::domain::schema::CoreEnvelopeVacuumDomain↩
Schema>
requires (Schema::template contains_domain<discretization::domain::StellarDomains>() && Schema::template contains_↩
domain<discretization::domain::VacuumDomain>() && Schema::template contains_boundary<discretization::domain::StellarSurfaceBoundary>()
&& Schema::template contains_boundary<discretization::domain::InfinitySurfaceBoundary>())
ReferenceDiscretization serif::geometry::deformation::backend::mfem::deformation_discretization (
const ::mfem::FiniteElementSpace & scalar_space,
const ::mfem::FiniteElementSpace & displacement_space,
::mfem::Mesh & logical_mesh) [nodiscard]

Resolve physical domain names through the schema rather than hard-coded MFEM attributes.

8.33.1.3 estimate_geometry_step_limit()

std::expected< GeometryStepLimit, DeformationFailure > serif::geometry::deformation::backend::mfem::estimate_geometry_↩
step_limit (
const mapping::DomainMapper & mapper,
const ::mfem::FiniteElementSpace & displacement_space,
const ::mfem::GridFunction & compactification_coordinate,
const ::mfem::Vector & accepted_displacement,
const ::mfem::Vector & displacement_direction,
std::span< const GeometrySampleRule > sample_rules,
GeometryStepOptions options = {}) [nodiscard]

Find the first sampled mapping boundary along accepted + step direction.

Inputs use the displacement space's true-DOF ordering. The compactification coordinate is fixed throughout this straight volume-displacement path. An exterior map must be affine in displacement. The result covers exactly the supplied rules, including repeats: quadrature rules need not be nested. Only serial spaces are supported. Parallel spaces are rejected before any mapping is evaluated.

8.33.1.4 make_deformation_computations()

template<typename Surface>
requires requires(const Surface& model, std::span<const std::array<double, 3>> positions) { { build_surface_embedding(model, positions) } -> std::same_as<std::expected<SurfaceEmbedding, DeformationFailure>>; }
auto serif::geometry::deformation::backend::mfem::make_deformation_computations () [nodiscard]

Compile fixed reference factors and evaluate shape motion with one dependency engine.

Note

Configuration values are named context inputs. Factories declare work; try_prepare performs it explicitly.

8.33.1.5 make_deformation_graph()

template<typename Surface>
auto serif::geometry::deformation::backend::mfem::make_deformation_graph () [nodiscard]

Provide a standalone graph while leaving the same computations available for equation composition.

8.33.1.6 make_deformation_inputs() [1/2]

template<typename Surface, typename Interior, typename Vacuum>
void serif::geometry::deformation::backend::mfem::make_deformation_inputs (
const DomainDeformation< Surface, Interior, Vacuum > & ,
const ReferenceDiscretization & ,
const std::vector< double > &&) [delete]

Reject temporary coefficient vectors whose storage would expire before preparation.

8.33.1.7 make_deformation_inputs() [2/2]

template<typename Surface, typename Interior, typename Vacuum>
auto serif::geometry::deformation::backend::mfem::make_deformation_inputs (
const DomainDeformation< Surface, Interior, Vacuum > & model,
const ReferenceDiscretization & discretization,
const std::vector< double > & coefficients) [nodiscard]

Bind physical choices and shape coordinates without evaluating deformation.

Note

Meshes, spaces, and the coefficient vector must outlive the context. Coordinate edits must notify MFEM through its normal node-update API.

8.33.1.8 make_surface_metric_computation()

auto serif::geometry::deformation::backend::mfem::make_surface_metric_computation () [inline], [nodiscard]

Optionally supply the reference surface mass diagonal in native shape coordinates.

Note

This reference metric is independent of shape coefficients and the volume extension; requesting it must not compile either.

8.33.1.9 make_volume_displacement_computations() [1/2]

template<discretization::domain::schema::IsSchema Schema>
void serif::geometry::deformation::backend::mfem::make_volume_displacement_computations (
const discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, Schema > &&
) [delete]

A graph cannot retain a displacement binding supplied as a temporary.

8.33.1.10 make_volume_displacement_computations() [2/2]

template<discretization::domain::schema::IsSchema Schema>
auto serif::geometry::deformation::backend::mfem::make_volume_displacement_computations (
const discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, Schema > &
binding) [nodiscard]

Connect a full component-block deformation vector to an equation's displacement binding.

Reference-dependent validation proves the two spaces assign the same meaning to each scalar coefficient. Subsequent computations own coefficients in the binding's global ordering and supported-DOF numbering, then return the existing CoefficientView into that immutable prerequisite. Cache result ownership keeps the vector's address stable throughout downstream evaluation and accepted-result reuse. The binding itself is borrowed and must outlive this graph and its results, as for volume equations.

8.33.1.11 sample_reference_positions()

std::expected< std::vector< std::array< double, 3 > >, DeformationFailure > serif::geometry::deformation::backend::mfem↩
::sample_reference_positions (
const ::mfem::FiniteElementSpace & scalar_space,
::mfem::Mesh & reference_mesh) [nodiscard]

Sample a paired mesh at the scalar companion nodes, checking shared-node agreement.

8.33.1.12 sample_reference_surface()

std::expected< ReferenceSurface, DeformationFailure > serif::geometry::deformation::backend::mfem::sample_reference_surface
(
const ::mfem::FiniteElementSpace & scalar_space,
int surface_attribute) [nodiscard]

Sample physical reference positions in ascending scalar-DOF order.

8.33.1.13 select_surface_scalar_dofs()

std::expected< std::vector< int >, DeformationFailure > serif::geometry::deformation::backend::mfem::select_surface_scalar↩
_dofs (
const ::mfem::FiniteElementSpace & scalar_space,
int surface_attribute) [nodiscard]

Select canonical surface nodes without retaining a mesh or finite-element-space pointer.

8.33.1.14 surface_parameter_mass_diagonal()

std::expected< std::vector< double >, DeformationFailure > serif::geometry::deformation::backend::mfem::surface_parameter↩
_mass_diagonal (
const ::mfem::FiniteElementSpace & scalar_space,
int surface_attribute,
const ReferenceSurface & reference_surface,
const SurfaceEmbedding & embedding) [nodiscard]

Measure each shape parameter with the reference boundary mass norm of its Cartesian finite-element trace.

Note

This is diag(S^T (M_boundary tensor I) S), independent of the interior/vacuum extension and of the current shape coefficients. It uses the mass integrator's standard quadrature and does not orthogonalize harmonic coordinates.

8.33.1.15 validate_companion_numbering()

DeformationStatus serif::geometry::deformation::backend::mfem::validate_companion_numbering (
const ::mfem::FiniteElementSpace & reference_space,
const ::mfem::FiniteElementSpace & companion_space) [nodiscard]

Require identical scalar coefficient meanings in two companion spaces. Vector component ordering may differ; element bases and their scalar-node indices must agree.

8.33.1.16 validate_scalar_reference_space()

DeformationStatus serif::geometry::deformation::backend::mfem::validate_scalar_reference_space (
const ::mfem::FiniteElementSpace & scalar_space) [nodiscard]

Reject spaces whose coefficients cannot be interpreted as serial nodal positions.

8.34 serif::geometry::deformation::backend::mfem::detail Namespace Reference

Classes

- struct LocatedBoundaryPoint

  Keep the best inverse boundary-map result and its physical residual.

Functions

- template<typename Value>
  std::expected< std::shared_ptr< const Value >, DeformationFailure > share_reference_result (std::expected< Value, DeformationFailure > result)

  Share a successfully compiled immutable table without copying it into each consumer.
- std::expected< RadialInterpolation, DeformationFailure > compile_radial_reference (const ReferenceDiscretization &input)

  Resolve changing mesh labels at preparation time, keeping material masks out of model constructors.
- double logical_radius (const ::mfem::Vector &position)

  Preserve the cubical logical radius used by the original radial foliation.
- std::expected< double, DeformationFailure > boundary_logical_radius (::mfem::Mesh &mesh, int attribute)

  Check a complete cubical shell and return its logical radius.
- std::expected< LocatedBoundaryPoint, DeformationFailure > locate_boundary_point (::mfem::Mesh &mesh, int attribute, const ::mfem::Vector &target)

  Locate a projected logical ray on a quadrilateral shell, preserving MeanField tolerances.

8.34.1 Function Documentation

8.34.1.1 boundary_logical_radius()

std::expected< double, DeformationFailure > serif::geometry::deformation::backend::mfem::detail::boundary_logical_radius (
::mfem::Mesh & mesh,
int attribute) [nodiscard]

Check a complete cubical shell and return its logical radius.

8.34.1.2 compile_radial_reference()

std::expected< RadialInterpolation, DeformationFailure > serif::geometry::deformation::backend::mfem::detail::compile_radial_reference (
const ReferenceDiscretization & input) [inline], [nodiscard]

Resolve changing mesh labels at preparation time, keeping material masks out of model constructors.

8.34.1.3 locate_boundary_point()

std::expected< LocatedBoundaryPoint, DeformationFailure > serif::geometry::deformation::backend::mfem::detail::locate_boundary_point (
::mfem::Mesh & mesh,
int attribute,
const ::mfem::Vector & target) [nodiscard]

Locate a projected logical ray on a quadrilateral shell, preserving MeanField tolerances.

8.34.1.4 logical_radius()

double serif::geometry::deformation::backend::mfem::detail::logical_radius (
const ::mfem::Vector & position) [nodiscard]

Preserve the cubical logical radius used by the original radial foliation.

8.34.1.5 share_reference_result()

template<typename Value>
std::expected< std::shared_ptr< const Value >, DeformationFailure > serif::geometry::deformation::backend::mfem::detail::share_reference_result (
std::expected< Value, DeformationFailure > result) [nodiscard]

Share a successfully compiled immutable table without copying it into each consumer.

8.35 serif::geometry::deformation::backend::mfem::volume_displacement_keys Namespace Reference

Classes

- struct CheckedBinding
- struct Coefficients

8.36 serif::geometry::deformation::detail Namespace Reference

Typedefs

- using DeterminantPolynomial = std::array<double, 4>
- using JacobianEntries = std::array<std::array<double, 3>, 3>

Functions

- DeformationStatus validate_disjoint_output (std::span< double > output, std::initializer_list< std::span< const double > > inputs)

  Preserve every action input until the last use, including nonlinear transpose derivatives. The list contains only the fixed number of mathematical arguments to an action; checking address intervals requires no allocation and never scans numerical coefficients.
- DeterminantPolynomial determinant_polynomial (const JacobianEntries &base, const JacobianEntries &direction, int dimension, double maximum_step, double determinant_floor) noexcept

  Expand det(base + t maximum_step direction) minus the floor for dimensions one through three.
- double evaluate_polynomial (const DeterminantPolynomial &coefficients, double parameter) noexcept

  Evaluate the cubic with fused operations, retaining small values near repeated roots.
- double first_boundary_parameter (const DeterminantPolynomial &coefficients) noexcept

  Find the first loss of strict positivity in (0, 1], including a tangent zero.

- std::expected< std::vector< SphericalHarmonicMode >, DeformationFailure > make_harmonic_modes (const SphericalHarmonicSurface &model, std::size_t node_count)

  Validate harmonic cutoffs before allocating a basis or enumerating modes.
- std::expected< std::vector< double >, DeformationFailure > sample_harmonic_basis (const SphericalHarmonicSurface &model, std::span< const SphericalHarmonicMode > modes, std::span< const std::array< double, 3 > > directions)

  Use normalized Legendre recurrence to avoid factorial growth while preserving the Condon–Shortley phase.
- DeformationStatus check_independent_harmonics (std::span< const double > basis, std::size_t node_count, std::size_t mode_count)

  Check sampled rank on a private copy, leaving physical coefficients and basis normalization unchanged.

8.36.1 Typedef Documentation

8.36.1.1 DeterminantPolynomial

using serif::geometry::deformation::detail::DeterminantPolynomial = std::array<double, 4>

8.36.1.2 JacobianEntries

using serif::geometry::deformation::detail::JacobianEntries = std::array<std::array<double, 3>, 3>

8.36.2 Function Documentation

8.36.2.1 check_independent_harmonics()

DeformationStatus serif::geometry::deformation::detail::check_independent_harmonics (
    std::span< const double > basis,
    std::size_t node_count,
    std::size_t mode_count) [nodiscard]

Check sampled rank on a private copy, leaving physical coefficients and basis normalization unchanged.

8.36.2.2 determinant_polynomial()

DeterminantPolynomial serif::geometry::deformation::detail::determinant_polynomial (
    const JacobianEntries & base,
    const JacobianEntries & direction,
    int dimension,
    double maximum_step,
    double determinant_floor) [nodiscard], [noexcept]

Expand det(base + t maximum_step direction) minus the floor for dimensions one through three.

8.36.2.3 evaluate_polynomial()

double serif::geometry::deformation::detail::evaluate_polynomial (
const DeterminantPolynomial & coefficients,
double parameter) [nodiscard], [noexcept]

Evaluate the cubic with fused operations, retaining small values near repeated roots.

8.36.2.4 first_boundary_parameter()

double serif::geometry::deformation::detail::first_boundary_parameter (
const DeterminantPolynomial & coefficients) [nodiscard], [noexcept]

Find the first loss of strict positivity in (0, 1], including a tangent zero.

Precondition

Coefficients are finite and the constant coefficient is strictly positive.

Returns

The boundary parameter, or infinity when the entire interval remains admissible.

8.36.2.5 make_harmonic_modes()

std::expected< std::vector< SphericalHarmonicMode >, DeformationFailure > serif::geometry::deformation::detail::make_↩
harmonic_modes (
const SphericalHarmonicSurface & model,
std::size_t node_count) [nodiscard]

Validate harmonic cutoffs before allocating a basis or enumerating modes.

8.36.2.6 sample_harmonic_basis()

std::expected< std::vector< double >, DeformationFailure > serif::geometry::deformation::detail::sample_harmonic_basis (
const SphericalHarmonicSurface & model,
std::span< const SphericalHarmonicMode > modes,
std::span< const std::array< double, 3 > > directions) [nodiscard]

Use normalized Legendre recurrence to avoid factorial growth while preserving the Condon–Shortley phase.

8.36.2.7 validate_disjoint_output()

DeformationStatus serif::geometry::deformation::detail::validate_disjoint_output (
std::span< double > output,
std::initializer_list< std::span< const double > > inputs) [inline], [nodiscard]

Preserve every action input until the last use, including nonlinear transpose derivatives. The list contains only the fixed number of mathematical arguments to an action; checking address intervals requires no allocation and never scans numerical coefficients.

8.37 serif::geometry::deformation::keys Namespace Reference

Classes

- struct DomainMap
- struct InteriorModel
- struct Linearization
- struct RadialInterpolation
- struct ReferenceDiscretization
- struct ReferenceSurface
- struct ShapeCoefficients
- struct SurfaceDisplacement
- struct SurfaceMap
- struct SurfaceMetric
- struct SurfaceModel
- struct VacuumModel
- struct VolumeDisplacement
- struct VolumeExtension

8.38 serif::geometry::deformation::spaces Namespace Reference

Classes

- struct ShapeParameters

  Independent coordinates belonging to the selected surface representation.
- struct SurfaceDisplacement

  Cartesian surface traces with components interleaved at each canonical node.
- struct VolumeDisplacement

  Displacement coefficients in the declared volume discretization.

8.39 serif::geometry::surface Namespace Reference

Namespaces

- namespace detail
- namespace keys

Classes

- struct BoundaryValues

  Values of one scalar term at an ordered collection of boundary points.
- struct DirectPressureSurfaceFormulation

  Enforce P_EOS(state) - P_surface = 0 without inventing an inverse EOS relation.
- class IsobaricSurface
- struct PressureSurfaceDescriptor
- struct PressureSurfaceFormulation

  The boundary residual is carrier minus the EOS value at prescribed pressure; no carrier quantity is privileged.
- struct SurfaceEOSResponse

  An EOS value and all of its local derivatives, evaluated together at one state.
- struct SurfaceFailure

  A rejected thermodynamic candidate, with a sample index when evaluating a boundary batch.

Concepts

- concept PressureSurfaceCompilable

  Whether the available state selects exactly one pressure-to-carrier EOS relation.
- concept DirectPressureSurfaceCompilable

  A direct pressure residual requires an explicitly chosen pressure relation and an input quantity whose equation row it replaces.
- concept PressureSurfaceFormulationType
- concept SurfaceScalarTerm

  Scalar thermodynamic data may come from any declared scalar space; boundary enforcement is a separate backend capability.
- concept SurfaceStateBindings

  Reuse dependency bindings, retaining distinct quantity roles even when they refer to one term.
- concept SurfaceResidualRow

  The quantity enforcing the boundary must have an actual equation row, not just sampled values.
- concept IsobaricSurfaceType

Typedefs

- template<typename EOS, typename Bindings, typename Carrier = void>
  using pressure_relation_t = std::tuple_element_t<0, typename detail::MatchingPressureRelations<std::remove_cvref_t<EOS>, std::remove_cvref_t<Bindings>, Carrier, typename std::remove_cvref_t<EOS>::Relations>::Matches>
- using Isobaric = IsobaricSurface
- template<PressureSurfaceFormulationType Formulation>
  using SurfacePoint = equations::EquationPoint<typename Formulation::Operands>
- template<PressureSurfaceFormulationType Formulation>
  using SurfaceLinearization = equations::PointLinearization<typename Formulation::Operands>

Enumerations

- enum class SurfaceFailureReason {
  non_finite_state , non_finite_eos_response , non_finite_linearization , inconsistent_sample_count , empty_surface }

Functions

- template<PressureSurfaceFormulationType Formulation>
  auto make_pressure_surface_graph ()

  Describe the isobaric calculation as an ordinary dependency graph, without evaluating the EOS.
- template<typename EOS, typename Bindings, typename Carrier = void>
  requires PressureSurfaceCompilable<EOS, Bindings, Carrier>
  auto make_pressure_surface_graph ()

  Infer a unique carrier-resolved formulation directly from an EOS and the available physical state.
- template<PressureSurfaceFormulationType Formulation>
  std::expected< SurfaceLinearization< Formulation >, SurfaceFailure > evaluate_pressure_surface (const typename Formulation::EquationOfState &equation_of_state, const IsobaricSurface &condition, const SurfacePoint< Formulation > &point)

  Evaluate the chosen isobaric residual and its analytical derivatives at one boundary point.
- template<SurfaceScalarTerm Term>
  auto boundary_values (std::vector< double > values)

  Supply owned boundary samples to an ordinary dependency context, keyed by their physical term.
- auto pressure_condition (IsobaricSurface condition)

  Supply the isobaric condition as an owned dependency source, with pressure tracked by value.

8.39.1 Typedef Documentation

8.39.1.1 Isobaric

using serif::geometry::surface::Isobaric = IsobaricSurface

8.39.1.2 pressure_relation_t

template<typename EOS, typename Bindings, typename Carrier = void>
using serif::geometry::surface::pressure_relation_t = std::tuple_element_t<0, typename detail::MatchingPressureRelations<std::remove_cvref_t<EOS>, std::remove_cvref_t<Bindings>, Carrier, typename std::remove_cvref_t<EOS>::Relations>::Matches>

8.39.1.3 SurfaceLinearization

template<PressureSurfaceFormulationType Formulation>
using serif::geometry::surface::SurfaceLinearization = equations::PointLinearization<typename Formulation::Operands>

8.39.1.4 SurfacePoint

template<PressureSurfaceFormulationType Formulation>
using serif::geometry::surface::SurfacePoint = equations::EquationPoint<typename Formulation::Operands>

8.39.2 Enumeration Type Documentation

8.39.2.1 SurfaceFailureReason

enum class serif::geometry::surface::SurfaceFailureReason [strong]

Enumerator

non_finite_state	
non_finite_eos_response	
non_finite_linearization	
inconsistent_sample_count	
empty_surface	

8.39.3 Function Documentation

8.39.3.1 boundary_values()

template<SurfaceScalarTerm Term>
auto serif::geometry::surface::boundary_values (
std::vector< double > values) [nodiscard]

Supply owned boundary samples to an ordinary dependency context, keyed by their physical term.

8.39.3.2 evaluate_pressure_surface()

```
template<PressureSurfaceFormulationType Formulation>
std::expected< SurfaceLinearization< Formulation >, SurfaceFailure > serif::geometry::surface::evaluate_pressure_surface (
        const typename Formulation::EquationOfState & equation_of_state,
        const IsobaricSurface & condition,
        const SurfacePoint< Formulation > & point)  [nodiscard]
```

Evaluate the chosen isobaric residual and its analytical derivatives at one boundary point.

Parameters

equation_of_state	The EOS declared by Formulation.
condition	The prescribed surface pressure, independent of the choice of carrier.
point	Values of the formulation's bound scalar terms, already sampled at a common boundary point.

Returns

A conventional equation linearization, or a numerical rejection. No preparation or borrowed state is retained.

Note

Domain exceptions from an existing throwing EOS propagate; no exception is used for surface validation.

8.39.3.3 make_pressure_surface_graph() [1/2]

```
template<PressureSurfaceFormulationType Formulation>
auto serif::geometry::surface::make_pressure_surface_graph ()  [nodiscard]
```

Describe the isobaric calculation as an ordinary dependency graph, without evaluating the EOS.

Template Parameters

Formulation	A carrier-resolved or explicitly selected direct-pressure surface formulation.

Returns

Two computations: the EOS response and the boundary residual with analytical derivatives.

Note

Bind keys::EquationOfState with a tracking policy covering all mutable EOS data, supply pressure_↩ condition and boundary_values, then use make_computation_cache<SurfaceFailure> and its normal try_prepare/get interface. The cache owns the complete preparation lifecycle.

8.39.3.4 make_pressure_surface_graph() [2/2]

template<typename EOS, typename Bindings, typename Carrier = void>
requires PressureSurfaceCompilable<EOS, Bindings, Carrier>
auto serif::geometry::surface::make_pressure_surface_graph () [nodiscard]

Infer a unique carrier-resolved formulation directly from an EOS and the available physical state.

8.39.3.5 pressure_condition()

auto serif::geometry::surface::pressure_condition (
IsobaricSurface condition) [inline], [nodiscard]

Supply the isobaric condition as an owned dependency source, with pressure tracked by value.

8.40 serif::geometry::surface::detail Namespace Reference

Classes

- struct AreSurfaceBindings
- struct AreSurfaceBindings< utils::dependencies::Bindings< Entries... > >
- struct BoundaryValueTracking
- struct MatchingPressureRelations
- struct MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Relations... > >
- struct PressureConditionTracking
- struct PressureSurfaceDescription
  Common physical metadata; numerical values and preparation belong to the dependency context and cache.
- struct RelationQuantities
- struct RelationQuantities< eos::relations::Relation< Output, Inputs... > >
- struct SurfaceOperands
- struct SurfaceOperands< utils::dependencies::KeyList< Terms... > >
- struct UniqueTerms
- struct UniqueTerms< utils::types::TypeList< Terms... > >

Typedefs

- template<typename Bindings, typename Quantities>
  using BoundTerms = typename UniqueTerms<utils::dependencies::ResolvedTargets<Bindings, Quantities>>::Type

Functions

- template<PressureSurfaceFormulationType Formulation, typename... Terms>
  auto make_surface_eos_computation (utils::dependencies::KeyList< Terms... >)

  Declare the exact EOS inputs from the formulation, so derivative metadata and cache dependencies cannot diverge.
- template<PressureSurfaceFormulationType Formulation>
  auto make_surface_linearization_computation ()

  Keep the inexpensive boundary residual separate from the reusable EOS response and its partials.
- template<typename... Terms>
  std::expected< std::size_t, SurfaceFailure > surface_sample_count (const BoundaryValues< Terms > &... columns)

  All EOS columns must describe the same ordered, nonempty set of boundary points.
- template<PressureSurfaceFormulationType Formulation, typename... Terms>
  SurfacePoint< Formulation > surface_point (const std::size_t sample, const BoundaryValues< Terms > &... columns)

  Assemble the conventional equation point from typed columns; no mesh assumptions enter the physical calculation.
- template<PressureSurfaceFormulationType Formulation, typename... Terms>
  std::expected< std::vector< SurfaceEOSResponse< Formulation > >, SurfaceFailure > evaluate_response_samples (const typename Formulation::EquationOfState &equation_of_state, const dimensions::PressureValue pressure, const BoundaryValues< Terms > &... columns)

  Evaluate a whole boundary batch in one graph node, preserving the first failed sample and publishing no partial batch.
- template<PressureSurfaceFormulationType Formulation>
  std::expected< std::vector< SurfaceLinearization< Formulation > >, SurfaceFailure > linearize_carrier_samples (const std::vector< SurfaceEOSResponse< Formulation > > &responses, const BoundaryValues< typename Formulation::CarrierTerm > &carrier)

  Broadcast only a genuinely pressure-only relation; otherwise every carrier sample needs its own EOS response.
- template<PressureSurfaceFormulationType Formulation>
  std::expected< std::vector< SurfaceLinearization< Formulation > >, SurfaceFailure > linearize_pressure_samples (const IsobaricSurface &condition, const std::vector< SurfaceEOSResponse< Formulation > > &responses)

  Changing the prescribed pressure of a direct-pressure residual reuses all EOS values and partials.
- template<typename Quantity, PressureSurfaceFormulationType Formulation>
  dimensions::QuantityValue< Quantity > state_value (const SurfacePoint< Formulation > &point)

  Read one thermodynamic role from its bound term, preserving the EOS argument's physical type.
- template<typename Quantity, PressureSurfaceFormulationType Formulation>
  dimensions::QuantityValue< Quantity > relation_input (const dimensions::PressureValue pressure, const SurfacePoint< Formulation > &point)

  Only the carrier-resolved formulation substitutes prescribed pressure for an EOS input.
- template<PressureSurfaceFormulationType Formulation, typename... Quantities>
  bool finite_state (const SurfacePoint< Formulation > &point, utils::dependencies::KeyList< Quantities... >)

  Validate precisely the state read by this calculation, leaving unrelated state outside its dependencies.
- template<typename Quantity, PressureSurfaceFormulationType Formulation, typename... Inputs>
  void accumulate_partial (const typename Formulation::EquationOfState &equation_of_state, const dimensions::PressureValue pressure, const SurfacePoint< Formulation > &point, SurfaceEOSResponse< Formulation > &response, dimensions::QuantityList< Inputs... >)

  Add each quantity's chain-rule contribution, including distinct roles bound to the same unknown.

- template<PressureSurfaceFormulationType Formulation, typename... Inputs, typename... StateQuantities>
  SurfaceEOSResponse< Formulation > evaluate_relation (const typename Formulation::EquationOf↩
  State &equation_of_state, const dimensions::PressureValue pressure, const SurfacePoint< Formula-
  tion > &point, dimensions::QuantityList< Inputs... > inputs, utils::dependencies::KeyList< State↩
  Quantities... >)

  Evaluate one selected relation and all required partials without any hidden EOS inversion.
- template<PressureSurfaceFormulationType Formulation>
  std::expected< SurfaceEOSResponse< Formulation >, SurfaceFailure > evaluate_surface_response
  (const typename Formulation::EquationOfState &equation_of_state, const dimensions::PressureValue
  pressure, const SurfacePoint< Formulation > &point)

  Reject non-finite thermodynamic inputs, EOS outputs, and partials as values.
- template<PressureSurfaceFormulationType Formulation>
  SurfaceLinearization< Formulation > linearize_surface (const dimensions::PressureValue pressure,
  const SurfacePoint< Formulation > &point, const SurfaceEOSResponse< Formulation > &response)

  Form the declared residual from an already evaluated EOS response, keeping the carrier identity explicit.
- template<PressureSurfaceFormulationType Formulation>
  std::expected< SurfaceLinearization< Formulation >, SurfaceFailure > checked_linearization
  (const dimensions::PressureValue pressure, const SurfacePoint< Formulation > &point, const
  SurfaceEOSResponse< Formulation > &response)

  Catch overflow in residual subtraction or derivative accumulation before publishing a linearization.
- template<typename EOS, typename Relation, typename... Quantities>
  consteval bool supports_partials (utils::dependencies::KeyList< Quantities... >)

  Check only the derivatives of unknown state; prescribed pressure has no Newton column.
- template<typename EOS, typename Bindings, typename Carrier, typename Relation>
  consteval bool matches_pressure_relation ()

  A resolved carrier must be supplied by one pressure-input relation, with all remaining inputs and partials available.

Variables

- template<typename Candidate>
  constexpr bool is_pressure_surface_formulation = false
- template<typename EOS, typename Bindings, typename Carrier>
  constexpr bool is_pressure_surface_formulation< PressureSurfaceFormulation< EOS, Bindings, Carrier > >
  = true
- template<typename EOS, typename Bindings, typename RowQuantity, typename Relation>
  constexpr bool is_pressure_surface_formulation< DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity,
  = true

8.40.1 Typedef Documentation

8.40.1.1 BoundTerms

template<typename Bindings, typename Quantities>
using serif::geometry::surface::detail::BoundTerms = typename UniqueTerms<utils::dependencies::ResolvedTargets<Bindings, Quantities>>::Type

8.40.2 Function Documentation

8.40.2.1 accumulate_partial()

```
template<typename Quantity, PressureSurfaceFormulationType Formulation, typename... Inputs>
void serif::geometry::surface::detail::accumulate_partial (
                const typename Formulation::EquationOfState & equation_of_state,
                const dimensions::PressureValue pressure,
                const SurfacePoint< Formulation > & point,
                SurfaceEOSResponse< Formulation > & response,
                dimensions::QuantityList< Inputs... > )
```

Add each quantity's chain-rule contribution, including distinct roles bound to the same unknown.

8.40.2.2 checked_linearization()

```
template<PressureSurfaceFormulationType Formulation>
std::expected< SurfaceLinearization< Formulation >, SurfaceFailure > serif::geometry::surface::detail::checked_linearization (
                const dimensions::PressureValue pressure,
                const SurfacePoint< Formulation > & point,
                const SurfaceEOSResponse< Formulation > & response)   [nodiscard]
```

Catch overflow in residual subtraction or derivative accumulation before publishing a linearization.

8.40.2.3 evaluate_relation()

```
template<PressureSurfaceFormulationType Formulation, typename... Inputs, typename... StateQuantities>
SurfaceEOSResponse< Formulation > serif::geometry::surface::detail::evaluate_relation (
                const typename Formulation::EquationOfState & equation_of_state,
                const dimensions::PressureValue pressure,
                const SurfacePoint< Formulation > & point,
                dimensions::QuantityList< Inputs... > inputs,
                utils::dependencies::KeyList< StateQuantities... > )   [nodiscard]
```

Evaluate one selected relation and all required partials without any hidden EOS inversion.

8.40.2.4 evaluate_response_samples()

```
template<PressureSurfaceFormulationType Formulation, typename... Terms>
std::expected< std::vector< SurfaceEOSResponse< Formulation > >, SurfaceFailure > serif::geometry::surface::detail::evaluate↩
_response_samples (
                const typename Formulation::EquationOfState & equation_of_state,
                const dimensions::PressureValue pressure,
                const BoundaryValues< Terms > &... columns)   [nodiscard]
```

Evaluate a whole boundary batch in one graph node, preserving the first failed sample and publishing no partial batch.

8.40.2.5 evaluate_surface_response()

```
template<PressureSurfaceFormulationType Formulation>
std::expected< SurfaceEOSResponse< Formulation >, SurfaceFailure > serif::geometry::surface::detail::evaluate_surface_↩
response (
        const typename Formulation::EquationOfState & equation_of_state,
        const dimensions::PressureValue pressure,
        const SurfacePoint< Formulation > & point)   [nodiscard]
```

Reject non-finite thermodynamic inputs, EOS outputs, and partials as values.

Note

Existing EOS implementations can still throw their documented domain errors; these propagate to the caller.

8.40.2.6 finite_state()

```
template<PressureSurfaceFormulationType Formulation, typename... Quantities>
bool serif::geometry::surface::detail::finite_state (
        const SurfacePoint< Formulation > & point,
        utils::dependencies::KeyList< Quantities... > )   [nodiscard]
```

Validate precisely the state read by this calculation, leaving unrelated state outside its dependencies.

8.40.2.7 linearize_carrier_samples()

```
template<PressureSurfaceFormulationType Formulation>
std::expected< std::vector< SurfaceLinearization< Formulation > >, SurfaceFailure > serif::geometry::surface::detail::linearize↩
_carrier_samples (
        const std::vector< SurfaceEOSResponse< Formulation > > & responses,
        const BoundaryValues< typename Formulation::CarrierTerm > & carrier)   [nodiscard]
```

Broadcast only a genuinely pressure-only relation; otherwise every carrier sample needs its own EOS response.

8.40.2.8 linearize_pressure_samples()

```
template<PressureSurfaceFormulationType Formulation>
std::expected< std::vector< SurfaceLinearization< Formulation > >, SurfaceFailure > serif::geometry::surface::detail::linearize↩
_pressure_samples (
        const IsobaricSurface & condition,
        const std::vector< SurfaceEOSResponse< Formulation > > & responses)   [nodiscard]
```

Changing the prescribed pressure of a direct-pressure residual reuses all EOS values and partials.

8.40.2.9 linearize_surface()

```
template<PressureSurfaceFormulationType Formulation>
SurfaceLinearization< Formulation > serif::geometry::surface::detail::linearize_surface (
                const dimensions::PressureValue pressure,
                const SurfacePoint< Formulation > & point,
                const SurfaceEOSResponse< Formulation > & response)   [nodiscard]
```

Form the declared residual from an already evaluated EOS response, keeping the carrier identity explicit.

8.40.2.10 make_surface_eos_computation()

```
template<PressureSurfaceFormulationType Formulation, typename... Terms>
auto serif::geometry::surface::detail::make_surface_eos_computation (
                utils::dependencies::KeyList< Terms... > )   [nodiscard]
```

Declare the exact EOS inputs from the formulation, so derivative metadata and cache dependencies cannot diverge.

8.40.2.11 make_surface_linearization_computation()

```
template<PressureSurfaceFormulationType Formulation>
auto serif::geometry::surface::detail::make_surface_linearization_computation ()   [nodiscard]
```

Keep the inexpensive boundary residual separate from the reusable EOS response and its partials.

8.40.2.12 matches_pressure_relation()

```
template<typename EOS, typename Bindings, typename Carrier, typename Relation>
bool serif::geometry::surface::detail::matches_pressure_relation ()   [nodiscard], [consteval]
```

A resolved carrier must be supplied by one pressure-input relation, with all remaining inputs and partials available.

8.40.2.13 relation_input()

```
template<typename Quantity, PressureSurfaceFormulationType Formulation>
dimensions::QuantityValue< Quantity > serif::geometry::surface::detail::relation_input (
                const dimensions::PressureValue pressure,
                const SurfacePoint< Formulation > & point)   [nodiscard]
```

Only the carrier-resolved formulation substitutes prescribed pressure for an EOS input.

8.40.2.14 state_value()

template<typename Quantity, PressureSurfaceFormulationType Formulation>
dimensions::QuantityValue< Quantity > serif::geometry::surface::detail::state_value (
const SurfacePoint< Formulation > & point) [nodiscard]

Read one thermodynamic role from its bound term, preserving the EOS argument's physical type.

8.40.2.15 supports_partials()

template<typename EOS, typename Relation, typename... Quantities>
bool serif::geometry::surface::detail::supports_partials (
utils::dependencies::KeyList< Quantities... >) [nodiscard], [consteval]

Check only the derivatives of unknown state; prescribed pressure has no Newton column.

8.40.2.16 surface_point()

template<PressureSurfaceFormulationType Formulation, typename... Terms>
SurfacePoint< Formulation > serif::geometry::surface::detail::surface_point (
const std::size_t sample,
const BoundaryValues< Terms > &... columns) [nodiscard]

Assemble the conventional equation point from typed columns; no mesh assumptions enter the physical calculation.

8.40.2.17 surface_sample_count()

template<typename... Terms>
std::expected< std::size_t, SurfaceFailure > serif::geometry::surface::detail::surface_sample_count (
const BoundaryValues< Terms > &... columns) [nodiscard]

All EOS columns must describe the same ordered, nonempty set of boundary points.

8.40.3 Variable Documentation

8.40.3.1 is_pressure_surface_formulation

template<typename Candidate>
bool serif::geometry::surface::detail::is_pressure_surface_formulation = false [inline], [constexpr]

8.40.3.2 is_pressure_surface_formulation< DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation > >

template<typename EOS, typename Bindings, typename RowQuantity, typename Relation>
bool serif::geometry::surface::detail::is_pressure_surface_formulation< DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation > > = true [inline], [constexpr]

8.40.3.3 is_pressure_surface_formulation< PressureSurfaceFormulation< EOS, Bindings, Carrier > >

template<typename EOS, typename Bindings, typename Carrier>
bool serif::geometry::surface::detail::is_pressure_surface_formulation< PressureSurfaceFormulation< EOS, Bindings, Carrier > > = true [inline], [constexpr]

8.41 serif::geometry::surface::keys Namespace Reference

Classes

- struct EOSResponse
- struct EquationOfState
- struct Linearization
- struct PressureCondition

8.42 serif::mapping Namespace Reference

Namespaces

- namespace compactification

Classes

- struct CompactificationPointData
- struct DMFixedMeshGeometry
- struct DMGeometryCache
- struct DMGeometryCacheStatistics
- struct DMReferenceDisplacementSample
- class DMWorkspace
- class DomainMapper
- class ElementCompactificationData
- class ElementDisplacementData
- struct ElementMappingData
- struct FaceMappingContext
- struct FaceMappingVariation
- struct FaceQuadratureContext
- class GridFunctionMappingEvaluator
- struct MappingPointContext
- struct MappingPointVariation
- class PhysicalPositionFunctionCoefficient
- class VolumeMappingCache
- struct VolumeMappingContext
- struct VolumeMappingVariation
- struct VolumeQuadratureContext

Enumerations

- enum class FaceElementSide : uint8_t { ELEMENT_1 , ELEMENT_2 }
- enum class MappingStatus : uint16_t {
  VALID , INVALID_DIMENSION = 400 , NON_FINITE_MAPPING_INPUT , INVALID_REFERENCE_RADIUS
  ,
  AT_COMPACTIFIED_INFINITY , OUTSIDE_REFERENCE_DOMAIN , NON_FINITE_MAPPING_RESULT
  , NON_POSITIVE_DETERMINATE }

Functions

- ElementDisplacementData element_displacement_data_from_element_vdofs (const mfem::Finite↩
  Element &element, mfem::Vector &displacement_dofs)
- std::unordered_map< MappingStatus, utils::errors::SERiFErrorCode > make_MappingStatus_SERiFErrorCode_map
  ()
- std::unordered_map< utils::errors::SERiFErrorCode, MappingStatus > make_SERIFErrorCode_MappingStatus_map
  ()
- void map_hdiv_flux_to_physical (const MappingPointContext &context, const mfem::Vector &reference_flux, mfem::Vector &physical_flux)
- void map_physical_flux_to_hdiv_reference (const MappingPointContext &context, const mfem::↩
  Vector &physical_flux, mfem::Vector &reference_flux)
- void map_reference_gradient_to_physical (const MappingPointContext &context, const mfem::Vector &reference_gradient, mfem::Vector &physical_gradient)
- void map_physical_gradient_to_reference (const MappingPointContext &context, const mfem::Vector &physical_gradient, mfem::Vector &reference_gradient)
- void map_reference_vector_gradient_to_physical (const MappingPointContext &context, const mfem::DenseMatrix &reference_vector_gradient, mfem::DenseMatrix &physical_vector_gradient)
- void map_physical_vector_gradient_to_reference (const MappingPointContext &context, const mfem::DenseMatrix &physical_vector_gradient, mfem::DenseMatrix &reference_vector_gradient)
- double map_hdiv_divergence_to_physical (const MappingPointContext &context, double reference↩
  _divergence)
- void compute_hdiv_mass_tensor (const MappingPointContext &context, mfem::DenseMatrix &mass_tensor)
- void compute_scalar_diffusion_tensor (const MappingPointContext &context, mfem::DenseMatrix &diffusion_tensor)
- void map_hcurl_field_to_physical (const MappingPointContext &context, const mfem::Vector &reference_field, mfem::Vector &physical_field)
- void map_physical_field_to_hcurl_reference (const MappingPointContext &context, const mfem::↩
  Vector &physical_field, mfem::Vector &reference_field)
- void map_hcurl_curl_to_physical (const MappingPointContext &context, const mfem::Vector &reference_curl, mfem::Vector &physical_curl)
- void map_physical_curl_to_hcurl_reference (const MappingPointContext &context, const mfem::↩
  Vector &physical_curl, mfem::Vector &reference_curl)
- void compute_hdiv_mass_tensor_variation (const MappingPointContext &context, const MappingPointVariation &variation, mfem::DenseMatrix &mass_tensor_variation)

Variables

- static auto MappingStatus_enum_string_map = meta::enum_to_enum_string_map<MappingStatus>()
- static auto MappingStatus_string_enum_map = meta::enum_to_string_enum_map<MappingStatus>()
- static auto MappingStatus_SERiFErrorCode_map = make_MappingStatus_SERiFErrorCode_map()
- static auto SERiFErrorCode_MappingStatus_map = make_SERIFErrorCode_MappingStatus_map()

8.42.1 Enumeration Type Documentation

8.42.1.1 FaceElementSide

enum class serif::mapping::FaceElementSide : uint8_t [strong]

Enumerator

ELEMENT_1	
ELEMENT_2	

8.42.1.2 MappingStatus

enum class serif::mapping::MappingStatus : uint16_t [strong]

Enumerator

VALID	
INVALID_DIMENSION	
NON_FINITE_MAPPING_INPUT	
INVALID_REFERENCE_RADIUS	
AT_COMPACTIFIED_INFINITY	
OUTSIDE_REFERENCE_DOMAIN	
NON_FINITE_MAPPING_RESULT	
NON_POSITIVE_DETERMINATE	

8.42.2 Function Documentation

8.42.2.1 compute_hdiv_mass_tensor()

void serif::mapping::compute_hdiv_mass_tensor (
const MappingPointContext & context,
mfem::DenseMatrix & mass_tensor)

8.42.2.2 compute_hdiv_mass_tensor_variation()

void serif::mapping::compute_hdiv_mass_tensor_variation (
const MappingPointContext & context,
const MappingPointVariation & variation,
mfem::DenseMatrix & mass_tensor_variation)

8.42.2.3 compute_scalar_diffusion_tensor()

void serif::mapping::compute_scalar_diffusion_tensor (
const MappingPointContext & context,
mfem::DenseMatrix & diffusion_tensor)

8.42.2.4 element_displacement_data_from_element_vdofs()

ElementDisplacementData serif::mapping::element_displacement_data_from_element_vdofs (
const mfem::FiniteElement & element,
mfem::Vector & displacement_dofs) [nodiscard]

8.42.2.5 make_MappingStatus_SERiFErrorCode_map()

std::unordered_map< MappingStatus, utils::errors::SERiFErrorCode > serif::mapping::make_MappingStatus_SERiFError↩
Code_map () [inline]

8.42.2.6 make_SERIFErrorCode_MappingStatus_map()

std::unordered_map< utils::errors::SERiFErrorCode, MappingStatus > serif::mapping::make_SERIFErrorCode_Mapping↩
Status_map () [inline]

8.42.2.7 map_hcurl_curl_to_physical()

void serif::mapping::map_hcurl_curl_to_physical (
const MappingPointContext & context,
const mfem::Vector & reference_curl,
mfem::Vector & physical_curl)

8.42.2.8 map_hcurl_field_to_physical()

void serif::mapping::map_hcurl_field_to_physical (
const MappingPointContext & context,
const mfem::Vector & reference_field,
mfem::Vector & physical_field)

8.42.2.9 map_hdiv_divergence_to_physical()

double serif::mapping::map_hdiv_divergence_to_physical (
const MappingPointContext & context,
double reference_divergence) [nodiscard]

8.42.2.10 map_hdiv_flux_to_physical()

```
void serif::mapping::map_hdiv_flux_to_physical (
        const MappingPointContext & context,
        const mfem::Vector & reference_flux,
        mfem::Vector & physical_flux)
```

8.42.2.11 map_physical_curl_to_hcurl_reference()

```
void serif::mapping::map_physical_curl_to_hcurl_reference (
        const MappingPointContext & context,
        const mfem::Vector & physical_curl,
        mfem::Vector & reference_curl)
```

8.42.2.12 map_physical_field_to_hcurl_reference()

```
void serif::mapping::map_physical_field_to_hcurl_reference (
        const MappingPointContext & context,
        const mfem::Vector & physical_field,
        mfem::Vector & reference_field)
```

8.42.2.13 map_physical_flux_to_hdiv_reference()

```
void serif::mapping::map_physical_flux_to_hdiv_reference (
        const MappingPointContext & context,
        const mfem::Vector & physical_flux,
        mfem::Vector & reference_flux)
```

8.42.2.14 map_physical_gradient_to_reference()

```
void serif::mapping::map_physical_gradient_to_reference (
        const MappingPointContext & context,
        const mfem::Vector & physical_gradient,
        mfem::Vector & reference_gradient)
```

8.42.2.15 map_physical_vector_gradient_to_reference()

```
void serif::mapping::map_physical_vector_gradient_to_reference (
        const MappingPointContext & context,
        const mfem::DenseMatrix & physical_vector_gradient,
        mfem::DenseMatrix & reference_vector_gradient)
```

8.42.2.16 map_reference_gradient_to_physical()

```
void serif::mapping::map_reference_gradient_to_physical (
        const MappingPointContext & context,
        const mfem::Vector & reference_gradient,
        mfem::Vector & physical_gradient)
```

8.42.2.17 map_reference_vector_gradient_to_physical()

```
void serif::mapping::map_reference_vector_gradient_to_physical (
        const MappingPointContext & context,
        const mfem::DenseMatrix & reference_vector_gradient,
        mfem::DenseMatrix & physical_vector_gradient)
```

8.42.3 Variable Documentation

8.42.3.1 MappingStatus_enum_string_map

auto serif::mapping::MappingStatus_enum_string_map = meta::enum_to_enum_string_map<MappingStatus>() [static]

8.42.3.2 MappingStatus_SERiFErrorCode_map

auto serif::mapping::MappingStatus_SERiFErrorCode_map = make_MappingStatus_SERiFErrorCode_map() [static]

8.42.3.3 MappingStatus_string_enum_map

auto serif::mapping::MappingStatus_string_enum_map = meta::enum_to_string_enum_map<MappingStatus>() [static]

8.42.3.4 SERiFErrorCode_MappingStatus_map

auto serif::mapping::SERiFErrorCode_MappingStatus_map = make_SERIFErrorCode_MappingStatus_map() [static]

8.43 serif::mapping::compactification Namespace Reference

Classes

- class ExteriorDomainMap
- struct ExteriorMapDirection
- struct ExteriorMapInput
- struct ExteriorMapResult
- struct ExteriorMapVariation
- class KelvinCompactification

8.44 serif::meta Namespace Reference

Functions

- template<typename EnumType>
  constexpr std::unordered_map< std::string_view, EnumType > enum_to_string_enum_map () noexcept
- template<typename EnumType>
  constexpr std::unordered_map< EnumType, std::string_view > enum_to_enum_string_map () noexcept
- template<typename EnumType>
  consteval std::string_view enum_to_string (EnumType value) noexcept
- template<typename T>
  consteval std::string_view type_name () noexcept
- template<typename T>
  consteval std::string_view template_name () noexcept

8.44.1 Function Documentation

8.44.1.1 enum_to_enum_string_map()

template<typename EnumType>
std::unordered_map< EnumType, std::string_view > serif::meta::enum_to_enum_string_map () [nodiscard], [constexpr], [noexcept]

8.44.1.2 enum_to_string()

template<typename EnumType>
std::string_view serif::meta::enum_to_string (
EnumType value) [nodiscard], [consteval], [noexcept]

8.44.1.3 enum_to_string_enum_map()

template<typename EnumType>
std::unordered_map< std::string_view, EnumType > serif::meta::enum_to_string_enum_map () [nodiscard], [constexpr], [noexcept]

8.44.1.4 template_name()

template<typename T>
std::string_view serif::meta::template_name () [nodiscard], [consteval], [noexcept]

8.44.1.5 type_name()

template<typename T>
std::string_view serif::meta::type_name () [nodiscard], [consteval], [noexcept]

8.45 serif::operators Namespace Reference

Namespaces

- namespace backend
- namespace detail

Classes

- struct BadContributionCollection
- struct BadContributionRow
- class BarotropicHydrostaticEquilibrium
- struct ContributionBinding

  Retain one leaf equation and the constant coefficient of its arithmetic expression.
- struct EquationPlan
- struct EquationPlan< EquationT, utils::types::TypeList< O... > >
- class EquationRegistry
- struct GradientRequest
- struct HasScalarVolumeOperands
- struct HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > >
- struct NestedBadCollection
- struct NodalRequest
- struct OperandValues
- struct VolumeContributions
- struct VolumeContributions< std::tuple< C... > >
- struct VolumeOperands
- struct VolumeOperands< C, utils::types::TypeList< O... > >

Concepts

- concept CanSum
- concept CanSubtract
- concept NodalValueEquation

  Nodal execution: an H1 value depending only on itself, with no explicit position dependence.
- concept ScalarValueTerm
- concept ScalarValueOperand
- concept ScalarVolumeEquation

  Equations executable by the current scalar-value volume operator.
- concept VolumeOperand

  Mathematical operand combinations implemented by the current volume backend.
- concept VolumeEquation

  One or more supported weak volume contributions assembled into a single residual row.

Typedefs

- using Closure = equations::BarotropicClosure<eos::models::Polytrope>
- using Density = discretization::blocks::density::mass
- using Enthalpy = discretization::blocks::enthalpy::specific
- using Schema = discretization::domain::schema::CoreEnvelopeVacuumDomainSchema
- using SurfacePressure = equations::PressureSurfaceConstraint<eos::models::Polytrope>
- using DensityValue = discretization::forms::Operand<Density>
- using EnthalpyValue = discretization::forms::Operand<Enthalpy>
- using EnthalpyGradient = discretization::forms::Operand<Enthalpy, discretization::forms::Operation::Gradient>
- using GravityValue = discretization::forms::Operand<discretization::blocks::gravity::gradient>
- using GravityDivergence = discretization::forms::Operand<discretization::blocks::gravity::gradient, discretization::forms::Operation::Divergence>
- using Pressure = equations::PressureForce<eos::models::Polytrope>
- using DisplacementDivergence = discretization::forms::Operand<discretization::blocks::displacement::vector, discretization::forms::Operation::Divergence>
- using Displacement = discretization::blocks::displacement::vector
- using DisplacementValue = discretization::forms::Operand<Displacement>
- using Potential = discretization::blocks::gravity::potential
- using MixedSamples = utils::types::TypeList<GravityValue, GravityDivergence, DensityValue>
- using Mechanical = equations::DisplacementForce<eos::models::Polytrope>
- using MechanicalDifference = equations::EquationDifference<Pressure, equations::EquationSum<equations::GravityFor equations::RotationalForce>>
- using ScaledMechanical = equations::EquationScale<MechanicalDifference>
- using ScaledPoisson = equations::EquationScale<equations::PoissonBalance>
- using Mass = equations::MassNormalization
- using Constant = discretization::blocks::barotropic_constant::mass_normalization
- using Omega = discretization::blocks::rotation::angular_velocity
- using Momentum = equations::AngularMomentum
- using CentralDensity = equations::CentralDensityConstraint<eos::models::Polytrope>

Functions

- template<equations::Equation EquationT>
  auto bind_contributions (std::shared_ptr< const EquationT > equation)

  Give each retained leaf a compile-time tuple position for independently cached contribution results.
- template<equations::Equation EquationT, typename FunctionT>
  void for_each_contribution (const EquationT &equation, FunctionT &&function)

8.45.1 Typedef Documentation

8.45.1.1 CentralDensity

using serif::operators::CentralDensity = equations::CentralDensityConstraint<eos::models::Polytrope>

8.45.1.2 Closure

using serif::operators::Closure = equations::BarotropicClosure<eos::models::Polytrope>

8.45.1.3 Constant

using serif::operators::Constant = discretization::blocks::barotropic_constant::mass_normalization

8.45.1.4 Density

using serif::operators::Density = discretization::blocks::density::mass

8.45.1.5 DensityValue

using serif::operators::DensityValue = discretization::forms::Operand<Density>

8.45.1.6 Displacement

using serif::operators::Displacement = discretization::blocks::displacement::vector

8.45.1.7 DisplacementDivergence

using serif::operators::DisplacementDivergence = discretization::forms::Operand<discretization::blocks::displacement::vector, discretization::forms::Operation::Divergence>

8.45.1.8 DisplacementValue

using serif::operators::DisplacementValue = discretization::forms::Operand<Displacement>

8.45.1.9 Enthalpy

using serif::operators::Enthalpy = discretization::blocks::enthalpy::specific

8.45.1.10 EnthalpyGradient

using serif::operators::EnthalpyGradient = discretization::forms::Operand<Enthalpy, discretization::forms::Operation::Gradient>

8.45.1.11 EnthalpyValue

using serif::operators::EnthalpyValue = discretization::forms::Operand<Enthalpy>

8.45.1.12 GravityDivergence

using serif::operators::GravityDivergence = discretization::forms::Operand<discretization::blocks::gravity::gradient, discretization::forms::Operation

8.45.1.13 GravityValue

using serif::operators::GravityValue = discretization::forms::Operand<discretization::blocks::gravity::gradient>

8.45.1.14 Mass

using serif::operators::Mass = equations::MassNormalization

8.45.1.15 Mechanical

using serif::operators::Mechanical = equations::DisplacementForce<eos::models::Polytrope>

8.45.1.16 MechanicalDifference

using serif::operators::MechanicalDifference = equations::EquationDifference<Pressure, equations::EquationSum<equations::GravityForce, equations::RotationalForce>>

8.45.1.17 MixedSamples

using serif::operators::MixedSamples = utils::types::TypeList<GravityValue, GravityDivergence, DensityValue>

8.45.1.18 Momentum

using serif::operators::Momentum = equations::AngularMomentum

8.45.1.19 Omega

using serif::operators::Omega = discretization::blocks::rotation::angular_velocity

8.45.1.20 Potential

using serif::operators::Potential = discretization::blocks::gravity::potential

8.45.1.21 Pressure

using serif::operators::Pressure = equations::PressureForce<eos::models::Polytrope>

8.45.1.22 ScaledMechanical

using serif::operators::ScaledMechanical = equations::EquationScale<MechanicalDifference>

8.45.1.23 ScaledPoisson

using serif::operators::ScaledPoisson = equations::EquationScale<equations::PoissonBalance>

8.45.1.24 Schema

using serif::operators::Schema = discretization::domain::schema::CoreEnvelopeVacuumDomainSchema

8.45.1.25 SurfacePressure

using serif::operators::SurfacePressure = equations::PressureSurfaceConstraint<eos::models::Polytrope>

8.45.2 Function Documentation

8.45.2.1 bind_contributions()

```
template<equations::Equation EquationT>
auto serif::operators::bind_contributions (
                std::shared_ptr< const EquationT > equation)   [nodiscard]
```

Give each retained leaf a compile-time tuple position for independently cached contribution results.

8.45.2.2 for_each_contribution()

```
template<equations::Equation EquationT, typename FunctionT>
void serif::operators::for_each_contribution (
                const EquationT & equation,
                FunctionT && function)
```

Visit leaves in declaration order, multiplying constant coefficients along each path. The callback receives (leaf equation, unique contribution index, coefficient).

8.46 serif::operators::backend Namespace Reference

Namespaces

- namespace mfem

8.47 serif::operators::backend::mfem Namespace Reference

Namespaces

- namespace nodal_keys
- namespace volume_keys

Classes

- struct BasisBindings
- class BasisEvaluation

  Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application.
- struct BasisWorkspace

  Reusable scratch storage for basis contractions, reducing repeated allocations.
- struct CoefficientTracking

  Track coefficient values using an owned snapshot and the identity of their binding.
- struct CoefficientView
- struct CompactificationCoefficients
- struct ContributionData
- struct DisplacementCoefficients
- struct ElementBasis

  Fixed-mesh basis samples; independent of the current coefficient values and displacement.
- struct ElementData

  Weighted equation data retains the exact basis, geometry, and samples used to compute it.
- struct ElementGeometry

  Physical mapping and mapped bases for one frozen displacement.
- class JacobianOperator
- struct MappingCoefficients
- struct MeshAndCoefficientBindings
- struct MetadataTracking

  Metadata is a value; moving its temporary wrapper does not change the mesh or bindings it describes.
- struct NodalData

  A frozen nodal residual and derivative retain the selection needed to interpret every entry.
- struct NodalDiscretization

  Fixed coefficient ordering and selected rows shared by the graph and its accepted results.
- struct NodalValueTracking

  Moving the context or copying equal coefficients does not invalidate the nodal calculation.
- struct NumericalWork

  Numerical work derived from the dependency engine's most recent report.
- struct OperandEvaluator
- struct OperandEvaluator< OperandT >

  Scalar values and H1/H(div) vector values or divergences share coefficient contractions.
- struct OperandSamples

  Reference samples remain available for derivatives through the physical mapping.
- struct PointBorderData

  Store the scalar and point linearization together so a result cannot mix accepted states.
- struct PointBorderDiscretization

  A point condition and its scalar border must use the same mesh.
- struct PreparationRejection
- struct SelectedNodalValues

  Selected coefficients are owned by the context; the selection has a stable shared identity.
- class TensorBasisCache

  Share immutable tensor plans within one operator; copied rules are keyed by their contents.
- struct VolumeConfiguration

Stable equation ownership also retains temporary and move-only contribution trees.

- struct VolumeData

  One accepted result contains ordered contribution batches and the assembled residual.
- struct VolumeDiscretization

  Borrowed discretization objects must outlive their graph and its accepted results.
- struct VolumeOptions

  Numerical choices fixed when the graph is constructed.
- struct VolumeScratch

  Scratch belongs to graph callables and is never a dependency source.
- struct Workspace

  Reusable scratch for applying the accepted element linearization.

Concepts

- concept ClosureResult
- concept ClosureInputsAccept
- concept AffinePointEquation

  An affine scalar point condition supplies the two off-diagonal border blocks.
- concept EvaluableOperand

  Check the complete interface used by preparation, Jacobian application, and its transpose.

Typedefs

- using Schema = discretization::domain::schema::CoreEnvelopeVacuumDomainSchema
- using Closure = equations::BarotropicClosure<eos::models::Polytrope>
- using Centering = equations::CenteringConstraint
- using Density = discretization::forms::Operand<discretization::blocks::density::mass>
- using DensityBinding = discretization::backend::mfem::TermBinding<Closure::Test, Schema>
- using EnthalpyBinding = discretization::backend::mfem::TermBinding<discretization::blocks::enthalpy::specific, Schema>
- using DisplacementBinding = discretization::backend::mfem::TermBinding<Centering::Test, Schema>
- using Vector = ::mfem::Vector
- using VolumeCoefficients
- using VolumeMesh = VolumeDiscretization<Closure::Test, Schema>
- using VolumeInputs
- using VolumeCache
- using NodalCoefficients
- using NodalMesh = decltype(make_point_discretization<Centering>(std::declval<const DisplacementBinding&>()))
- using NodalCache
- using PreparationMode = utils::dependencies::PreparationMode
- template<std::size_t Contribution, typename OperandT>
  using VolumeOperandResult = std::conditional_t<OperandEvaluator<OperandT>::geometry_dependent, volume_keys::PhysicalValues<Contribution, OperandT>, volume_keys::ReferenceValues<Contribution, OperandT>>

  Values need a mapping node only when the operand's physical representation depends on geometry.
- template<typename CacheT>
  using VolumeEquationOf = typename std::remove_cvref_t<decltype(std::declval<const CacheT&>().template get<volume_keys::Linearization>())>::Equation

Recover the equation type from a cache's accepted result type, without accessing a result.

- template<typename ElementT>
  using ElementBatch = std::vector<std::shared_ptr<const ElementT>>

  A stage owns immutable element results, shared with downstream stages and accepted linearizations.
- template<typename OperandT>
  using SamplesFor = OperandSamples

  Associate an operand type pack with equally ordered sample arguments.

Enumerations

- enum class PreparationRejectionReason { MAPPING , EQUATION , NON_FINITE }
- enum class ContractionMode

  Options enum to control how the tensor contractions are performed.
- enum class ReferenceOperation

  Options enum to control what is actually evaluated at the reference points.
- enum class ContractionMode

  Options enum to control how the tensor contractions are performed.

Functions

- template<typename ContributionT, std::size_t Count>
  void element_residual (const ElementData< Count > &data, ::mfem::Vector &result, BasisWorkspace &workspace)

  Integrate the cached scalar/vector physics against its actual test operation.
- template<VolumeEquation EquationT, typename ContributionT>
  auto geometry_integrand_variation (const ElementData< EquationPlan< EquationT >::operand_count > &data, int point, const mapping::VolumeMappingContext &context, const mapping::VolumeMappingVariation &variation)

  Geometry-only variation of the weighted physics; shared by forward and transpose actions.
- template<VolumeEquation EquationT, typename ContributionT>
  void element_jacobian_action (const ElementData< EquationPlan< EquationT >::operand_count > &data, const std::array< const discretization::backend::mfem::DofMap *, EquationPlan< EquationT >::column_count > &columns, const std::array< const ::mfem::Vector *, EquationPlan< EquationT >::column_count > &directions, const discretization::backend::mfem::DofMap &displacement_binding, const ::mfem::Vector &displacement_direction, const ::mfem::FiniteElementSpace *compactification_space, const mapping::DomainMapper &mapper, Workspace &workspace, ::mfem::Vector &result)

  Differentiate weight, physics, and both operand and test mapping rules in one path.
- template<VolumeEquation EquationT, typename ContributionT>
  ElementBasis< EquationPlan< EquationT >::operand_count > sample_element_basis (const ContributionT &equation, int element, const discretization::backend::mfem::DofMap &row, const discretization::backend::mfem::DofMap &displacement_binding, const std::array< const discretization::backend::mfem::DofMap *, EquationPlan< EquationT >::column_count > &columns, const mapping::DomainMapper &mapper, const discretization::quadrature::Policy &policy, TensorBasisCache &basis_cache, ContractionMode contraction_mode)

  Sample the contribution's quadrature rule and bases on the fixed computational mesh.
- input binding element_values (basis.element, input.values, workspace.element_values)

- template<VolumeEquation EquationT, typename ContributionT>
  std::expected< ElementGeometry< EquationPlan< EquationT >::operand_count >, PreparationRejection > map_element_geometry (const ElementBasis< EquationPlan< EquationT >::operand_count > &basis, const discretization::backend::mfem::DofMap &displacement_binding, const ::mfem::Vector &displacement, const MappingCoefficients &compactification, const mapping::DomainMapper &mapper, Workspace &workspace)

  Freeze displacement and compactification, then map quadrature points and bases to physical space.
- template<EvaluableOperand OperandT, std::size_t Count>
  ::mfem::Vector map_element_operand (const ::mfem::Vector &reference, const ElementGeometry< Count > &geometry, Workspace &workspace)

  Apply an operand's physical transformation to its cached fixed-mesh samples.
- template<VolumeEquation EquationT, typename ContributionT>
  equations::EquationPoint< typename ContributionT::Operands > element_equation_point (const std::array< std::shared_ptr< const ::mfem::Vector >, EquationPlan< EquationT >::operand_count > &physical_values, const mapping::MappingPointContext &context, int point)

  Pack the physical samples requested by this contribution at one quadrature point.
- template<VolumeEquation EquationT, typename ContributionT>
  bool store_element_linearization (const equations::PointLinearization< typename ContributionT::↩ Operands, typename OperandEvaluator< equations::TestOperandOf< ContributionT > >::Value > &physics, int point, double weight, ElementData< EquationPlan< EquationT >::operand_count > &data)

  Weight a point's residual and derivatives; reject any non-finite stored component.
- template<VolumeEquation EquationT, typename ContributionT>
  std::expected< ElementData< EquationPlan< EquationT >::operand_count >, PreparationRejection > evaluate_element_equation (const ContributionT &equation, double coefficient, std::shared_ptr< const ElementBasis< EquationPlan< EquationT >::operand_count > > basis, std::shared_ptr< const ElementGeometry< EquationPlan< EquationT >::operand_count > > geometry, std::array< std↩ ::shared_ptr< const ::mfem::Vector >, EquationPlan< EquationT >::operand_count > reference_↩ values, const std::array< std::shared_ptr< const ::mfem::Vector >, EquationPlan< EquationT >↩ ::operand_count > &physical_values, Workspace &workspace)

  Evaluate and weight the equation while retaining its immutable preparation inputs.
- template<VolumeEquation EquationT, typename ContributionT>
  void element_jacobian_transpose (const ElementData< EquationPlan< EquationT >::operand_count > &data, const discretization::backend::mfem::DofMap &row, const discretization::backend::mfem::DofMap &displacement_binding, const ::mfem::Vector &residual_dual, const ::mfem::FiniteElementSpace &compactification_space, const mapping::DomainMapper &mapper, Workspace &workspace, std::array<::mfem::Vector, EquationPlan< EquationT >::column_count > &column_duals, ::mfem::Vector &displacement_dual)
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  auto make_nodal_equation_computation (EquationT equation, NodalDiscretization< EquationT, SchemaT > discretization)

  Declare the point or boundary calculation; its fixed equation and selection move with the graph.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  auto make_nodal_equation_cache (EquationT equation, const NodalDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  Construct an ordinary dependency cache without evaluating the nodal equation.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  auto make_point_border_cache (EquationT equation, const PointBorderDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  Keep the border scalar and point linearization in one dependency graph and one accepted result.

- template<typename CacheT, NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  std::expected< void, PreparationRejection > try_prepare_nodal (CacheT &cache, const NodalDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients, const PreparationMode mode=PreparationMode::reuse_unchanged)

  Validate and gather the selected coefficients, then let the dependency cache decide what needs evaluation.
- template<typename CacheT, AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  std::expected< void, PreparationRejection > try_prepare_point_border (CacheT &cache, const PointBorderDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_↩ coefficients, const PreparationMode mode=PreparationMode::reuse_unchanged)

  Validate both point and scalar sources before publishing the complete border calculation.
- template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
  void prepare_nodal (CacheT &cache, const DiscretizationT &discretization, const CoefficientsT &term_coefficients, const PreparationMode mode=PreparationMode::reuse_unchanged)

  Translate a numerical rejection at the caller-facing throwing boundary.
- template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
  void prepare_point_border (CacheT &cache, const DiscretizationT &discretization, const CoefficientsT &term_coefficients, const PreparationMode mode=PreparationMode::reuse_unchanged)

  Translate an affine border's numerical rejection at the caller-facing throwing boundary.
- template<typename CacheT>
  NumericalWork nodal_work_report (const CacheT &cache)

  Derive numerical work from the dependency report, without a second report lifecycle.
- template<NodalValueEquation EquationT>
  equations::EquationPoint< typename EquationT::Operands > make_nodal_equation_point (const ::mfem::Vector &values, const int first)

  Convert one scalar node or one complete vector point into the equation's declared input.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  std::expected< void, PreparationRejection > store_nodal_linearization (const equations::PointLinearization< typename EquationT::Operands, discretization::forms::OperandValue< equations::TestOperandOf< EquationT > > > &physics, const int first, NodalData< EquationT, SchemaT > &data)

  Accept only finite, position-independent linearizations before exposing their row data.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  std::expected< NodalData< EquationT, SchemaT >, PreparationRejection > evaluate_nodal_values (const EquationT &equation, const NodalDiscretization< EquationT, SchemaT > &discretization, const ::mfem::Vector &values)

  Evaluate the equation once per selected point, keeping residuals and derivatives in one cached result.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void validate_nodal_selection (const NodalDiscretization< EquationT, SchemaT > &discretization)

  Check stored selection invariants before dereferencing or gathering any coefficient.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void validate_nodal_discretization (const NodalDiscretization< EquationT, SchemaT > &discretization)

  Admit only nodal spaces and the complete selection required by the equation's point or boundary support.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  auto make_nodal_discretization (const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > &residual_binding, discretization::backend::mfem::DofSelection selected_rows)

  Validate a caller-supplied selection immediately, before constructing its computation graph.

- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  requires requires { EquationT::reference_position; }
  auto make_point_discretization (const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > &residual_binding)

  Select the equation's physical point once; coordinate changes require a new selection.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  requires (discretization::domain::IsBoundary<equations::SupportOf<EquationT>> && discretization::blocks::SpatialTermTraits<typename EquationT::Test>::rank == 0)
  auto make_boundary_discretization (const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > &residual_binding)

  Select the scalar equation's boundary through its declared schema support.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  auto make_point_border_discretization (const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > &residual_binding, const discretization::backend::mfem::TermBinding< typename EquationT::Border, SchemaT > &border_binding)

  Couple the selected scalar point to a global scalar on the same mesh.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  SelectedNodalValues select_nodal_values (const NodalDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  Validate the whole supplied vector, but cache only coefficients selected by this equation.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  auto make_nodal_dependency_context (const NodalDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  The test term keys the selected coefficients needed by a nodal equation.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  auto make_point_border_dependency_context (const PointBorderDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  A border scalar is a separate tracked source; changing it does not reevaluate the point equation.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void validate_nodal_output (const NodalData< EquationT, SchemaT > &data, const ::mfem::Vector &result)

  A frozen linearization still requires a current selection and a correctly sized output.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void apply_residual_rows (const NodalData< EquationT, SchemaT > &data, ::mfem::Vector &result)

  Replace only selected entries of an assembled residual with the accepted nodal residual.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void apply_jacobian_rows (const NodalData< EquationT, SchemaT > &data, const ::mfem::Vector &direction, ::mfem::Vector &result)

  Replace complete Jacobian rows, including every unconstrained coupling at the selected rows.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  requires requires(const CoefficientsT& direction) { direction.template get<typename EquationT::Test>(); }
  void apply_jacobian_rows (const NodalData< EquationT, SchemaT > &data, const CoefficientsT &direction, ::mfem::Vector &result)

  Select the nodal direction by its physical term and verify its coefficient ordering.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void mask_residual_dual (const NodalData< EquationT, SchemaT > &data, const ::mfem::Vector &dual, ::mfem::Vector &volume_dual)

  Remove constrained rows before applying the volume Jacobian transpose.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void add_jacobian_transpose_rows (const NodalData< EquationT, SchemaT > &data, const ::mfem::Vector &dual, ::mfem::Vector &result)

Add the constrained-row transpose after applying the volume transpose to the masked dual.

- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void validate_point_border_outputs (const PointBorderData< EquationT, SchemaT > &data, const ::mfem::Vector &carrier, const ::mfem::Vector &phase)

  Point borders accept one carrier vector and one global scalar output.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void add_point_border_residual (const PointBorderData< EquationT, SchemaT > &data, ::mfem::Vector &carrier, ::mfem::Vector &phase)

  Add the accepted border force to the carrier and set the scalar phase residual.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  void apply_point_border_jacobian (const PointBorderData< EquationT, SchemaT > &data, const CoefficientsT &direction, ::mfem::Vector &carrier, ::mfem::Vector &phase)

  Apply the two off-diagonal Jacobian blocks of an affine scalar point border.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void apply_point_border_jacobian_transpose (const PointBorderData< EquationT, SchemaT > &data, const ::mfem::Vector &carrier_dual, const ::mfem::Vector &phase_dual, ::mfem::Vector &carrier_result, ::mfem::Vector &border_result)

  Add the coordinate transpose of both border blocks to their coefficient duals.
- template<typename CacheT, typename PrepareT>
  std::expected< void, PreparationRejection > try_prepare_validated (CacheT &cache, PrepareT &&prepare)

  A failed validation or calculation must make the previous accepted result inaccessible.
- template<VolumeEquation RowEquationT, typename ContributionT>
  const ::mfem::IntegrationRule & equation_rule (const ContributionT &equation, const discretization::quadrature::Policy &policy, ::mfem::ElementTransformation &transformation, const ::mfem::FiniteElement &displacement↩
  _element, discretization::quadrature::MappingKind mapping_kind=discretization::quadrature::MappingKind::general)

  Resolve one rule covering all operand forms and the geometry weight.
- template<typename BindingT>
  CoefficientView< BindingT > bind (const BindingT &binding, const ::mfem::Vector &values)

  Borrow a term binding and its coefficients without extending their lifetimes.
- template<typename BindingT>
  void bind (const BindingT &, ::mfem::Vector &&)=delete
- template<typename BindingT>
  void bind (const BindingT &, const ::mfem::Vector &&)=delete
- template<typename BindingT>
  void bind (const BindingT &&, const ::mfem::Vector &)=delete
- template<typename... BindingTs>
  auto make_term_coefficients (BindingTs... bindings)

  Associate each physical term with its bound coefficients in an ordinary dependency context.
- template<VolumeEquation EquationT, typename SchemaT>
  void validate_volume_data (const VolumeData< EquationT, SchemaT > &data)

  Reject changes to discretization objects borrowed by an accepted linearization.
- template<VolumeEquation EquationT, typename SchemaT, typename VisitT>
  void for_each_volume_element (const VolumeData< EquationT, SchemaT > &data, VisitT &&visit)

  Visit each contribution's own batch in equation order, without scanning other contributions.
- template<VolumeEquation EquationT, typename SchemaT>
  int volume_height (const VolumeData< EquationT, SchemaT > &data)

  Return the supported residual size.
- template<VolumeEquation EquationT, typename SchemaT>
  int volume_width (const VolumeData< EquationT, SchemaT > &data)

Packed directions contain unique coefficient terms followed by displacement.

- template<VolumeEquation EquationT, typename SchemaT>
  std::size_t tensor_element_count (const VolumeData< EquationT, SchemaT > &data)

  Count contribution-elements using tensor contractions for any sampled basis.
- template<VolumeEquation EquationT, typename SchemaT>
  int integration_order (const VolumeData< EquationT, SchemaT > &data, int element)

  Return the largest contribution quadrature order on a supported mesh element.
- template<VolumeEquation EquationT, typename SchemaT>
  void volume_residual (const VolumeData< EquationT, SchemaT > &data, ::mfem::Vector &result)

  Copy the residual already assembled and checked when this result was accepted.
- template<typename RowBindingT>
  void accumulate_element_result (const RowBindingT &residual_binding, int element, Workspace &workspace, utils::CompensatedSum &global_sum)

  Global rows accumulate with compensation; spatial rows scatter into their supported DOFs.
- template<VolumeEquation EquationT, typename SchemaT>
  void apply_volume_direction (const VolumeData< EquationT, SchemaT > &data, const std↩ ::array< const ::mfem::Vector *, EquationPlan< EquationT >::column_count > &directions, const ::mfem::Vector &displacement_direction, Workspace &workspace, ::mfem::Vector &result)

  Apply the element linearizations and assemble a supported residual direction.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  void apply_jacobian (const VolumeData< EquationT, SchemaT > &data, const CoefficientsT &direction, const ::mfem::Vector &displacement_direction, Workspace &workspace, ::mfem::Vector &result)

  Apply a direction whose terms explicitly name the accepted coefficient bindings.
- template<VolumeEquation EquationT, typename SchemaT>
  void apply_jacobian (const VolumeData< EquationT, SchemaT > &data, const ::mfem::Vector &direction, Workspace &workspace, ::mfem::Vector &result)

  Unpack unique coefficient terms and displacement, then use the same numerical action.
- template<VolumeEquation EquationT, typename SchemaT>
  void apply_jacobian_transpose (const VolumeData< EquationT, SchemaT > &data, const ::mfem::Vector &residual_dual, Workspace &workspace, ::mfem::Vector &result)

  Assemble the full coordinate transpose in the same packed order as the forward action.
- template<std::size_t Contribution, typename OperandT, VolumeEquation EquationT, typename SchemaT>
  auto make_operand_computations (const std::shared_ptr< VolumeScratch > &scratch)

  Declare one coefficient contraction, followed by a physical mapping when the space requires it.
- template<std::size_t Index, VolumeEquation EquationT, typename SchemaT, typename ContributionT, typename... Operands>
  auto make_contribution_computations (const std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > &configuration, const std::shared_ptr< VolumeScratch > &scratch, ContributionBinding< ContributionT > contribution, utils::types::TypeList< Operands... >)

  Build one contribution's branch in the same order as its numerical description.
- template<VolumeEquation EquationT, typename SchemaT, std::size_t... Indices>
  auto connect_volume_contributions (const std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > &configuration, std::index_sequence< Indices... >)

  Expose contribution computations so deformation and equations can share one dependency graph.
- template<VolumeEquation EquationT, std::size_t Index, typename CacheT>
  void report_contribution_preparation (const CacheT &cache, NumericalWork &report)

  Convert the dependency engine's node statuses to numerical work counts for one contribution.
- template<VolumeEquation EquationT, typename SchemaT>
  auto make_volume_equation_computations (EquationT equation, const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, VolumeOptions options={})

Declare reusable volume computations without selecting their source of displacement.

- template<VolumeEquation EquationT, typename SchemaT, typename InputsT>
  auto make_volume_equation_cache (EquationT equation, const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const InputsT &inputs, VolumeOptions options={})

  Construct a standalone cache from the same computations used by a coupled deformation graph.
- template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
  std::expected< void, PreparationRejection > try_prepare_volume (CacheT &cache, const DiscretizationT &discretization, const CoefficientsT &term_coefficients, const ::mfem::Vector &displacement_↩ coefficients, PreparationMode mode=PreparationMode::reuse_unchanged)

  Validate current inputs and atomically publish a complete volume linearization.
- template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
  void prepare_volume (CacheT &cache, const DiscretizationT &discretization, const CoefficientsT &term_coefficients, const ::mfem::Vector &displacement_coefficients, PreparationMode mode=PreparationMode::reuse_unchanged)

  Throw a caller-visible numerical error when a complete preparation cannot be accepted.
- template<typename CacheT>
  NumericalWork volume_work_report (const CacheT &cache)

  Derive work counts from accepted node statuses; no second report lifecycle is stored.
- template<typename TestT, typename SchemaT>
  VolumeDiscretization (const discretization::backend::mfem::TermBinding< TestT, SchemaT > &, const discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, SchemaT > &, const ::mfem::GridFunction &, const mapping::DomainMapper &) -> VolumeDiscretization< TestT, SchemaT >
- template<typename TestT, typename SchemaT>
  void validate_volume_discretization (const VolumeDiscretization< TestT, SchemaT > &discretization)

  Check borrowed mesh objects before sampling or applying accepted results.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT, typename... Columns>
  auto make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const CoefficientsT &term_coefficients, utils::types::TypeList< Columns... >)

  Supply precisely the consumed terms; unrelated coefficients are not graph dependencies.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  auto make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  Bind equation and reference inputs when displacement will be produced inside the dependency graph.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT, typename... Columns>
  auto make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const CoefficientsT &term_coefficients, const ::mfem::Vector &displacement_↩ coefficients, utils::types::TypeList< Columns... > columns)

  Append a checked external displacement while preserving each original resource's tracking policy.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  auto make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const CoefficientsT &term_coefficients, const ::mfem::Vector &displacement_↩ coefficients)

  Build a checked source context without retaining the coefficient vectors.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  void make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &, const CoefficientsT &, ::mfem::Vector &&)=delete
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  void make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &, const CoefficientsT &, const ::mfem::Vector &&)=delete

- template<VolumeEquation EquationT, typename SchemaT, typename ContributionT>
  auto sample_contribution_basis (const VolumeConfiguration< EquationT, SchemaT > &configuration, const ContributionBinding< ContributionT > &contribution, const BasisBindings< EquationPlan< EquationT >::column_count > &bindings, TensorBasisCache &cache)

  Sample the reference discretization once for each supported element of this contribution.
- template<VolumeEquation EquationT, typename OperandT, typename BindingT>
  OperandSamples sample_contribution_operand (const ElementBatch< ElementBasis< EquationPlan< EquationT >::operand_count > > &basis, const CoefficientView< BindingT > &input, Workspace &workspace)

  Only this operand's coefficients and reference basis determine its reference samples.
- template<VolumeEquation EquationT, typename SchemaT, typename ContributionT>
  std::expected< ElementBatch< ElementGeometry< EquationPlan< EquationT >::operand_count > >, PreparationRejection > map_contribution_geometry (const VolumeConfiguration< EquationT, SchemaT > &configuration, const ElementBatch< ElementBasis< EquationPlan< EquationT >::operand_count > > &basis, const CoefficientView< typename VolumeConfiguration< EquationT, SchemaT >::DisplacementBinding > &displacement, const MappingCoefficients &compactification, Workspace &workspace)

  Freeze mapping data and mapped bases independently of the equation's coefficient inputs.
- template<typename OperandT, std::size_t OperandCount>
  OperandSamples map_contribution_operand (const OperandSamples &samples, const ElementBatch< ElementGeometry< OperandCount > > &geometry, Workspace &workspace)

  Map only operands whose physical representation depends on deformation.
- template<VolumeEquation EquationT, typename ContributionT, typename... Operands>
  std::expected< ContributionData< EquationPlan< EquationT >::operand_count >, PreparationRejection > evaluate_contribution_equation (const ContributionBinding< ContributionT > &contribution, const ElementBatch< ElementBasis< EquationPlan< EquationT >::operand_count > > &basis, const ElementBatch< ElementGeometry< EquationPlan< EquationT >::operand_count > > &geometry, Workspace &workspace, const SamplesFor< Operands > &... samples)

  Evaluate one equation contribution; its basis, mapping and individual operand samples are already prepared.
- template<VolumeEquation EquationT, typename SchemaT, typename... Contributions>
  std::expected< VolumeData< EquationT, SchemaT >, PreparationRejection > collect_volume_results (const std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > &configuration, const BasisBindings< EquationPlan< EquationT >::column_count > &bindings, Workspace &workspace, const Contributions &... contributions)

  Assemble once, retaining contribution order and compensated global scalar accumulation.

Variables

- template<VolumeEquation EquationT, EvaluableOperand OperandT, typename BindingT>
  constexpr std::size_t index = equations::OperandValues<typename Plan::Operands>::template index<OperandT>()

  Contract one operand's fixed-mesh basis with its current coefficients.
- ::mfem::Vector values

8.47.1 Typedef Documentation

8.47.1.1 Centering

using serif::operators::backend::mfem::Centering = equations::CenteringConstraint

8.47.1.2 Closure

using serif::operators::backend::mfem::Closure = equations::BarotropicClosure<eos::models::Polytrope>

8.47.1.3 Density

using serif::operators::backend::mfem::Density = discretization::forms::Operand<discretization::blocks::density::mass>

8.47.1.4 DensityBinding

using serif::operators::backend::mfem::DensityBinding = discretization::backend::mfem::TermBinding<Closure::Test, Schema>

8.47.1.5 DisplacementBinding

using serif::operators::backend::mfem::DisplacementBinding = discretization::backend::mfem::TermBinding<Centering::Test, Schema>

8.47.1.6 ElementBatch

template<typename ElementT>
using serif::operators::backend::mfem::ElementBatch = std::vector<std::shared_ptr<const ElementT>>

A stage owns immutable element results, shared with downstream stages and accepted linearizations.

8.47.1.7 EnthalpyBinding

using serif::operators::backend::mfem::EnthalpyBinding = discretization::backend::mfem::TermBinding<discretization::blocks::enthalpy::specific, Schema>

8.47.1.8 NodalCache

using serif::operators::backend::mfem::NodalCache

Initial value:

```
decltype(make_nodal_equation_cache(
      std::declval<Centering>(), std::declval<NodalMesh>(), std::declval<const NodalCoefficients&>()
  ))
```

8.47.1.9 NodalCoefficients

using serif::operators::backend::mfem::NodalCoefficients

Initial value:

```
decltype(make_term_coefficients(
      bind(std::declval<const DisplacementBinding&>(), std::declval<const Vector&>())
  ))
```

8.47.1.10 NodalMesh

using serif::operators::backend::mfem::NodalMesh = decltype(make_point_discretization<Centering>(std::declval<const DisplacementBinding&>()))

8.47.1.11 PreparationMode

using serif::operators::backend::mfem::PreparationMode = utils::dependencies::PreparationMode

8.47.1.12 SamplesFor

template<typename OperandT>
using serif::operators::backend::mfem::SamplesFor = OperandSamples

Associate an operand type pack with equally ordered sample arguments.

8.47.1.13 Schema

using serif::operators::backend::mfem::Schema = discretization::domain::schema::CoreEnvelopeVacuumDomainSchema

8.47.1.14 Vector

using serif::operators::backend::mfem::Vector = ::mfem::Vector

8.47.1.15 VolumeCache

using serif::operators::backend::mfem::VolumeCache

Initial value:

```
decltype(make_volume_equation_cache(
      std::declval<Closure>(), std::declval<VolumeMesh>(), std::declval<const VolumeInputs&>()
  ))
```

8.47.1.16 VolumeCoefficients

using serif::operators::backend::mfem::VolumeCoefficients

Initial value:

```
decltype(make_term_coefficients(
      bind(std::declval<const DensityBinding&>(), std::declval<const Vector&>()),
      bind(std::declval<const EnthalpyBinding&>(), std::declval<const Vector&>())
  ))
```

8.47.1.17 VolumeEquationOf

template<typename CacheT>
using serif::operators::backend::mfem::VolumeEquationOf = typename std::remove_cvref_t<decltype(std::declval<const CacheT&>().template get<volume_keys::Linearization>())>::Equation

Recover the equation type from a cache's accepted result type, without accessing a result.

8.47.1.18 VolumeInputs

using serif::operators::backend::mfem::VolumeInputs

Initial value:

```
decltype(make_volume_inputs<Closure>(
      std::declval<const VolumeMesh&>(), std::declval<const VolumeCoefficients&>(), std::declval<const Vector&>()
  ))
```

8.47.1.19 VolumeMesh

using serif::operators::backend::mfem::VolumeMesh = VolumeDiscretization<Closure::Test, Schema>

8.47.1.20 VolumeOperandResult

template<std::size_t Contribution, typename OperandT>
using serif::operators::backend::mfem::VolumeOperandResult = std::conditional_t<OperandEvaluator<OperandT>::geometry_dependent, volume_keys::PhysicalValues<Contribution, OperandT>, volume_keys::ReferenceValues<Contribution, OperandT>>

Values need a mapping node only when the operand's physical representation depends on geometry.

8.47.2 Enumeration Type Documentation

8.47.2.1 ContractionMode [1/2]

enum class serif::discretization::backend::mfem::ContractionMode [strong]

Options enum to control how the tensor contractions are performed.

8.47.2.2 ContractionMode [2/2]

enum class serif::discretization::backend::mfem::ContractionMode [strong]

Options enum to control how the tensor contractions are performed.

8.47.2.3 PreparationRejectionReason

enum class serif::operators::backend::mfem::PreparationRejectionReason [strong]

Enumerator

MAPPING	
EQUATION	
NON_FINITE	

8.47.2.4 ReferenceOperation

enum class serif::discretization::backend::mfem::ReferenceOperation [strong]

Options enum to control what is actually evaluated at the reference points.

8.47.3 Function Documentation

8.47.3.1 accumulate_element_result()

```
template<typename RowBindingT>
void serif::operators::backend::mfem::accumulate_element_result (
                const RowBindingT & residual_binding,
                int element,
                Workspace & workspace,
                utils::CompensatedSum & global_sum)
```

Global rows accumulate with compensation; spatial rows scatter into their supported DOFs.

8.47.3.2 add_jacobian_transpose_rows()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::add_jacobian_transpose_rows (
                const NodalData< EquationT, SchemaT > & data,
                const ::mfem::Vector & dual,
                ::mfem::Vector & result)
```

Add the constrained-row transpose after applying the volume transpose to the masked dual.

8.47.3.3 add_point_border_residual()

```
template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::add_point_border_residual (
                const PointBorderData< EquationT, SchemaT > & data,
                ::mfem::Vector & carrier,
                ::mfem::Vector & phase)
```

Add the accepted border force to the carrier and set the scalar phase residual.

8.47.3.4 apply_jacobian() [1/2]

```
template<VolumeEquation EquationT, typename SchemaT>
void serif::operators::backend::mfem::apply_jacobian (
                const VolumeData< EquationT, SchemaT > & data,
                const ::mfem::Vector & direction,
                Workspace & workspace,
                ::mfem::Vector & result)
```

Unpack unique coefficient terms and displacement, then use the same numerical action.

8.47.3.5 apply_jacobian() [2/2]

```
template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
void serif::operators::backend::mfem::apply_jacobian (
                const VolumeData< EquationT, SchemaT > & data,
                const CoefficientsT & direction,
                const ::mfem::Vector & displacement_direction,
                Workspace & workspace,
                ::mfem::Vector & result)
```

Apply a direction whose terms explicitly name the accepted coefficient bindings.

8.47.3.6 apply_jacobian_rows() [1/2]

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::apply_jacobian_rows (
                const NodalData< EquationT, SchemaT > & data,
                const ::mfem::Vector & direction,
                ::mfem::Vector & result)
```

Replace complete Jacobian rows, including every unconstrained coupling at the selected rows.

8.47.3.7 apply_jacobian_rows() [2/2]

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
requires requires(const CoefficientsT& direction) { direction.template get<typename EquationT::Test>(); }
void serif::operators::backend::mfem::apply_jacobian_rows (
                const NodalData< EquationT, SchemaT > & data,
                const CoefficientsT & direction,
                ::mfem::Vector & result)
```

Select the nodal direction by its physical term and verify its coefficient ordering.

8.47.3.8 apply_jacobian_transpose()

```
template<VolumeEquation EquationT, typename SchemaT>
void serif::operators::backend::mfem::apply_jacobian_transpose (
                const VolumeData< EquationT, SchemaT > & data,
                const ::mfem::Vector & residual_dual,
                Workspace & workspace,
                ::mfem::Vector & result)
```

Assemble the full coordinate transpose in the same packed order as the forward action.

8.47.3.9 apply_point_border_jacobian()

```
template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
void serif::operators::backend::mfem::apply_point_border_jacobian (
                const PointBorderData< EquationT, SchemaT > & data,
                const CoefficientsT & direction,
                ::mfem::Vector & carrier,
                ::mfem::Vector & phase)
```

Apply the two off-diagonal Jacobian blocks of an affine scalar point border.

8.47.3.10 apply_point_border_jacobian_transpose()

```
template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::apply_point_border_jacobian_transpose (
                const PointBorderData< EquationT, SchemaT > & data,
                const ::mfem::Vector & carrier_dual,
                const ::mfem::Vector & phase_dual,
                ::mfem::Vector & carrier_result,
                ::mfem::Vector & border_result)
```

Add the coordinate transpose of both border blocks to their coefficient duals.

8.47.3.11 apply_residual_rows()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::apply_residual_rows (
                const NodalData< EquationT, SchemaT > & data,
                ::mfem::Vector & result)
```

Replace only selected entries of an assembled residual with the accepted nodal residual.

8.47.3.12 apply_volume_direction()

```
template<VolumeEquation EquationT, typename SchemaT>
void serif::operators::backend::mfem::apply_volume_direction (
                const VolumeData< EquationT, SchemaT > & data,
                const std::array< const ::mfem::Vector *, EquationPlan< EquationT >::column_count > & directions,
                const ::mfem::Vector & displacement_direction,
                Workspace & workspace,
                ::mfem::Vector & result)
```

Apply the element linearizations and assemble a supported residual direction.

8.47.3.13 bind() [1/4]

```
template<typename BindingT>
void serif::operators::backend::mfem::bind (
                const BindingT && ,
                const ::mfem::Vector & )   [delete]
```

8.47.3.14 bind() [2/4]

```
template<typename BindingT>
void serif::operators::backend::mfem::bind (
                const BindingT & ,
                ::mfem::Vector && )   [delete]
```

8.47.3.15 bind() [3/4]

```
template<typename BindingT>
void serif::operators::backend::mfem::bind (
                const BindingT & ,
                const ::mfem::Vector && )   [delete]
```

8.47.3.16 bind() [4/4]

```
template<typename BindingT>
CoefficientView< BindingT > serif::operators::backend::mfem::bind (
                const BindingT & binding,
                const ::mfem::Vector & values)   [nodiscard]
```

Borrow a term binding and its coefficients without extending their lifetimes.

8.47.3.17 collect_volume_results()

```
template<VolumeEquation EquationT, typename SchemaT, typename... Contributions>
std::expected< VolumeData< EquationT, SchemaT >, PreparationRejection > serif::operators::backend::mfem::collect_↩
volume_results (
        const std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > & configuration,
        const BasisBindings< EquationPlan< EquationT >::column_count > & bindings,
        Workspace & workspace,
        const Contributions &... contributions)  [nodiscard]
```

Assemble once, retaining contribution order and compensated global scalar accumulation.

8.47.3.18 connect_volume_contributions()

```
template<VolumeEquation EquationT, typename SchemaT, std::size_t... Indices>
auto serif::operators::backend::mfem::connect_volume_contributions (
        const std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > & configuration,
        std::index_sequence< Indices... > )  [nodiscard]
```

Expose contribution computations so deformation and equations can share one dependency graph.

8.47.3.19 element_equation_point()

```
template<VolumeEquation EquationT, typename ContributionT>
equations::EquationPoint< typename ContributionT::Operands > serif::operators::backend::mfem::element_equation_point (
        const std::array< std::shared_ptr< const ::mfem::Vector >, EquationPlan< EquationT >::operand_count > &
physical_values,
        const mapping::MappingPointContext & context,
        int point)  [nodiscard]
```

Pack the physical samples requested by this contribution at one quadrature point.

8.47.3.20 element_jacobian_action()

```
template<VolumeEquation EquationT, typename ContributionT>
void serif::operators::backend::mfem::element_jacobian_action (
        const ElementData< EquationPlan< EquationT >::operand_count > & data,
        const std::array< const discretization::backend::mfem::DofMap *, EquationPlan< EquationT >::column_count
> & columns,
        const std::array< const ::mfem::Vector *, EquationPlan< EquationT >::column_count > & directions,
        const discretization::backend::mfem::DofMap & displacement_binding,
        const ::mfem::Vector & displacement_direction,
        const ::mfem::FiniteElementSpace * compactification_space,
        const mapping::DomainMapper & mapper,
        Workspace & workspace,
        ::mfem::Vector & result)
```

Differentiate weight, physics, and both operand and test mapping rules in one path.

8.47.3.21 element_jacobian_transpose()

```
template<VolumeEquation EquationT, typename ContributionT>
void serif::operators::backend::mfem::element_jacobian_transpose (
                const ElementData< EquationPlan< EquationT >::operand_count > & data,
                const discretization::backend::mfem::DofMap & row,
                const discretization::backend::mfem::DofMap & displacement_binding,
                const ::mfem::Vector & residual_dual,
                const ::mfem::FiniteElementSpace & compactification_space,
                const mapping::DomainMapper & mapper,
                Workspace & workspace,
                std::array<::mfem::Vector, EquationPlan< EquationT >::column_count > & column_duals,
                ::mfem::Vector & displacement_dual)
```

Pull a residual dual back through test sampling, local physics, operand sampling and geometry. Geometry has twelve local inputs (three values and nine canonical derivatives), independent of FE order.

8.47.3.22 element_residual()

```
template<typename ContributionT, std::size_t Count>
void serif::operators::backend::mfem::element_residual (
                const ElementData< Count > & data,
                ::mfem::Vector & result,
                BasisWorkspace & workspace)
```

Integrate the cached scalar/vector physics against its actual test operation.

8.47.3.23 element_values()

```
input binding serif::operators::backend::mfem::element_values (
                basis. element,
                input. values,
                workspace. element_values)
```

8.47.3.24 equation_rule()

```
template<VolumeEquation RowEquationT, typename ContributionT>
const ::mfem::IntegrationRule & serif::operators::backend::mfem::equation_rule (
                const ContributionT & equation,
                const discretization::quadrature::Policy & policy,
                ::mfem::ElementTransformation & transformation,
                const ::mfem::FiniteElement & displacement_element,
                discretization::quadrature::MappingKind mapping_kind = discretization::quadrature::MappingKind::general)
[nodiscard]
```

Resolve one rule covering all operand forms and the geometry weight.

8.47.3.25 evaluate_contribution_equation()

```
template<VolumeEquation EquationT, typename ContributionT, typename... Operands>
std::expected< ContributionData< EquationPlan< EquationT >::operand_count >, PreparationRejection > serif::operators↩
::backend::mfem::evaluate_contribution_equation (
        const ContributionBinding< ContributionT > & contribution,
        const ElementBatch< ElementBasis< EquationPlan< EquationT >::operand_count > > & basis,
        const ElementBatch< ElementGeometry< EquationPlan< EquationT >::operand_count > > & geometry,
        Workspace & workspace,
        const SamplesFor< Operands > &... samples)  [nodiscard]
```

Evaluate one equation contribution; its basis, mapping and individual operand samples are already prepared.

8.47.3.26 evaluate_element_equation()

```
template<VolumeEquation EquationT, typename ContributionT>
std::expected< ElementData< EquationPlan< EquationT >::operand_count >, PreparationRejection > serif::operators↩
::backend::mfem::evaluate_element_equation (
        const ContributionT & equation,
        double coefficient,
        std::shared_ptr< const ElementBasis< EquationPlan< EquationT >::operand_count > > basis,
        std::shared_ptr< const ElementGeometry< EquationPlan< EquationT >::operand_count > > geometry,
         std::array< std::shared_ptr< const ::mfem::Vector >, EquationPlan< EquationT >::operand_count >
reference_values,
        const std::array< std::shared_ptr< const ::mfem::Vector >, EquationPlan< EquationT >::operand_count > &
physical_values,
        Workspace & workspace)  [nodiscard]
```

Evaluate and weight the equation while retaining its immutable preparation inputs.

8.47.3.27 evaluate_nodal_values()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
std::expected< NodalData< EquationT, SchemaT >, PreparationRejection > serif::operators::backend::mfem::evaluate_nodal↩
_values (
        const EquationT & equation,
        const NodalDiscretization< EquationT, SchemaT > & discretization,
        const ::mfem::Vector & values)  [nodiscard]
```

Evaluate the equation once per selected point, keeping residuals and derivatives in one cached result.

8.47.3.28 for_each_volume_element()

```
template<VolumeEquation EquationT, typename SchemaT, typename VisitT>
void serif::operators::backend::mfem::for_each_volume_element (
        const VolumeData< EquationT, SchemaT > & data,
        VisitT && visit)
```

Visit each contribution's own batch in equation order, without scanning other contributions.

8.47.3.29 geometry_integrand_variation()

```
template<VolumeEquation EquationT, typename ContributionT>
auto serif::operators::backend::mfem::geometry_integrand_variation (
                const ElementData< EquationPlan< EquationT >::operand_count > & data,
                int point,
                const mapping::VolumeMappingContext & context,
                const mapping::VolumeMappingVariation & variation)   [nodiscard]
```

Geometry-only variation of the weighted physics; shared by forward and transpose actions.

8.47.3.30 integration_order()

```
template<VolumeEquation EquationT, typename SchemaT>
int serif::operators::backend::mfem::integration_order (
                const VolumeData< EquationT, SchemaT > & data,
                int element)   [nodiscard]
```

Return the largest contribution quadrature order on a supported mesh element.

8.47.3.31 make_boundary_discretization()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
requires (discretization::domain::IsBoundary<equations::SupportOf<EquationT>> && discretization::blocks::SpatialTermTraits<typename
EquationT::Test>::rank == 0)
auto serif::operators::backend::mfem::make_boundary_discretization (
                const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > & residual_↩
binding)   [nodiscard]
```

Select the scalar equation's boundary through its declared schema support.

8.47.3.32 make_contribution_computations()

```
template<std::size_t Index, VolumeEquation EquationT, typename SchemaT, typename ContributionT, typename...
Operands>
auto serif::operators::backend::mfem::make_contribution_computations (
                const std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > & configuration,
                const std::shared_ptr< VolumeScratch > & scratch,
                ContributionBinding< ContributionT > contribution,
                utils::types::TypeList< Operands... > )   [nodiscard]
```

Build one contribution's branch in the same order as its numerical description.

8.47.3.33 make_nodal_dependency_context()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
auto serif::operators::backend::mfem::make_nodal_dependency_context (
                const NodalDiscretization< EquationT, SchemaT > & discretization,
                const CoefficientsT & term_coefficients)   [nodiscard]
```

The test term keys the selected coefficients needed by a nodal equation.

8.47.3.34 make_nodal_discretization()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
auto serif::operators::backend::mfem::make_nodal_discretization (
        const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > & residual_↩
binding,
        discretization::backend::mfem::DofSelection selected_rows)  [nodiscard]
```

Validate a caller-supplied selection immediately, before constructing its computation graph.

8.47.3.35 make_nodal_equation_cache()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
auto serif::operators::backend::mfem::make_nodal_equation_cache (
        EquationT equation,
        const NodalDiscretization< EquationT, SchemaT > & discretization,
        const CoefficientsT & term_coefficients)  [nodiscard]
```

Construct an ordinary dependency cache without evaluating the nodal equation.

8.47.3.36 make_nodal_equation_computation()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
auto serif::operators::backend::mfem::make_nodal_equation_computation (
        EquationT equation,
        NodalDiscretization< EquationT, SchemaT > discretization)  [nodiscard]
```

Declare the point or boundary calculation; its fixed equation and selection move with the graph.

8.47.3.37 make_nodal_equation_point()

```
template<NodalValueEquation EquationT>
equations::EquationPoint< typename EquationT::Operands > serif::operators::backend::mfem::make_nodal_equation_point (
        const ::mfem::Vector & values,
        const int first)  [nodiscard]
```

Convert one scalar node or one complete vector point into the equation's declared input.

8.47.3.38 make_operand_computations()

```
template<std::size_t Contribution, typename OperandT, VolumeEquation EquationT, typename SchemaT>
auto serif::operators::backend::mfem::make_operand_computations (
        const std::shared_ptr< VolumeScratch > & scratch)  [nodiscard]
```

Declare one coefficient contraction, followed by a physical mapping when the space requires it.

8.47.3.39 make_point_border_cache()

```
template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
auto serif::operators::backend::mfem::make_point_border_cache (
                EquationT equation,
                const PointBorderDiscretization< EquationT, SchemaT > & discretization,
                const CoefficientsT & term_coefficients)   [nodiscard]
```

Keep the border scalar and point linearization in one dependency graph and one accepted result.

8.47.3.40 make_point_border_dependency_context()

```
template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
auto serif::operators::backend::mfem::make_point_border_dependency_context (
                const PointBorderDiscretization< EquationT, SchemaT > & discretization,
                const CoefficientsT & term_coefficients)   [nodiscard]
```

A border scalar is a separate tracked source; changing it does not reevaluate the point equation.

8.47.3.41 make_point_border_discretization()

```
template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
auto serif::operators::backend::mfem::make_point_border_discretization (
                const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > & residual_↩
binding,
                const discretization::backend::mfem::TermBinding< typename EquationT::Border, SchemaT > & border_↩
binding)   [nodiscard]
```

Couple the selected scalar point to a global scalar on the same mesh.

8.47.3.42 make_point_discretization()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
requires requires { EquationT::reference_position; }
auto serif::operators::backend::mfem::make_point_discretization (
                const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > & residual_↩
binding)   [nodiscard]
```

Select the equation's physical point once; coordinate changes require a new selection.

8.47.3.43 make_term_coefficients()

```
template<typename... BindingTs>
auto serif::operators::backend::mfem::make_term_coefficients (
                BindingTs... bindings)   [nodiscard]
```

Associate each physical term with its bound coefficients in an ordinary dependency context.

8.47.3.44 make_volume_equation_cache()

```
template<VolumeEquation EquationT, typename SchemaT, typename InputsT>
auto serif::operators::backend::mfem::make_volume_equation_cache (
                EquationT equation,
                const VolumeDiscretization< typename EquationT::Test, SchemaT > & discretization,
                const InputsT & inputs,
                VolumeOptions options = {})   [nodiscard]
```

Construct a standalone cache from the same computations used by a coupled deformation graph.

8.47.3.45 make_volume_equation_computations()

```
template<VolumeEquation EquationT, typename SchemaT>
auto serif::operators::backend::mfem::make_volume_equation_computations (
                EquationT equation,
                const VolumeDiscretization< typename EquationT::Test, SchemaT > & discretization,
                VolumeOptions options = {})   [nodiscard]
```

Declare reusable volume computations without selecting their source of displacement.

Note

This allocates scratch and owns the equation configuration; numerical work begins at try_prepare. Borrowed discretization objects must outlive the computations and their accepted results.

8.47.3.46 make_volume_inputs() [1/6]

```
template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
void serif::operators::backend::mfem::make_volume_inputs (
                const VolumeDiscretization< typename EquationT::Test, SchemaT > & ,
                const CoefficientsT & ,
                ::mfem::Vector && )   [delete]
```

8.47.3.47 make_volume_inputs() [2/6]

```
template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
void serif::operators::backend::mfem::make_volume_inputs (
                const VolumeDiscretization< typename EquationT::Test, SchemaT > & ,
                const CoefficientsT & ,
                const ::mfem::Vector && )   [delete]
```

8.47.3.48 make_volume_inputs() [3/6]

```
template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
auto serif::operators::backend::mfem::make_volume_inputs (
                const VolumeDiscretization< typename EquationT::Test, SchemaT > & discretization,
                const CoefficientsT & term_coefficients)   [nodiscard]
```

Bind equation and reference inputs when displacement will be produced inside the dependency graph.

8.47.3.49 make_volume_inputs() [4/6]

```
template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
auto serif::operators::backend::mfem::make_volume_inputs (
                const VolumeDiscretization< typename EquationT::Test, SchemaT > & discretization,
                const CoefficientsT & term_coefficients,
                const ::mfem::Vector & displacement_coefficients)   [nodiscard]
```

Build a checked source context without retaining the coefficient vectors.

8.47.3.50 make_volume_inputs() [5/6]

```
template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT, typename... Columns>
auto serif::operators::backend::mfem::make_volume_inputs (
                const VolumeDiscretization< typename EquationT::Test, SchemaT > & discretization,
                const CoefficientsT & term_coefficients,
                const ::mfem::Vector & displacement_coefficients,
                utils::types::TypeList< Columns... > columns)   [nodiscard]
```

Append a checked external displacement while preserving each original resource's tracking policy.

8.47.3.51 make_volume_inputs() [6/6]

```
template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT, typename... Columns>
auto serif::operators::backend::mfem::make_volume_inputs (
                const VolumeDiscretization< typename EquationT::Test, SchemaT > & discretization,
                const CoefficientsT & term_coefficients,
                utils::types::TypeList< Columns... > )   [nodiscard]
```

Supply precisely the consumed terms; unrelated coefficients are not graph dependencies.

8.47.3.52 map_contribution_geometry()

```
template<VolumeEquation EquationT, typename SchemaT, typename ContributionT>
std::expected< ElementBatch< ElementGeometry< EquationPlan< EquationT >::operand_count > >, PreparationRejection
> serif::operators::backend::mfem::map_contribution_geometry (
                const VolumeConfiguration< EquationT, SchemaT > & configuration,
                const ElementBatch< ElementBasis< EquationPlan< EquationT >::operand_count > > & basis,
                const CoefficientView< typename VolumeConfiguration< EquationT, SchemaT >::DisplacementBinding > &
displacement,
                const MappingCoefficients & compactification,
                Workspace & workspace)   [nodiscard]
```

Freeze mapping data and mapped bases independently of the equation's coefficient inputs.

8.47.3.53 map_contribution_operand()

```
template<typename OperandT, std::size_t OperandCount>
OperandSamples serif::operators::backend::mfem::map_contribution_operand (
                const OperandSamples & samples,
                const ElementBatch< ElementGeometry< OperandCount > > & geometry,
                Workspace & workspace)   [nodiscard]
```

Map only operands whose physical representation depends on deformation.

8.47.3.54 map_element_geometry()

```
template<VolumeEquation EquationT, typename ContributionT>
std::expected< ElementGeometry< EquationPlan< EquationT >::operand_count >, PreparationRejection > serif::operators↩
::backend::mfem::map_element_geometry (
                const ElementBasis< EquationPlan< EquationT >::operand_count > & basis,
                const discretization::backend::mfem::DofMap & displacement_binding,
                const ::mfem::Vector & displacement,
                const MappingCoefficients & compactification,
                const mapping::DomainMapper & mapper,
                Workspace & workspace)   [nodiscard]
```

Freeze displacement and compactification, then map quadrature points and bases to physical space.

8.47.3.55 map_element_operand()

```
template<EvaluableOperand OperandT, std::size_t Count>
::mfem::Vector serif::operators::backend::mfem::map_element_operand (
                const ::mfem::Vector & reference,
                const ElementGeometry< Count > & geometry,
                Workspace & workspace)   [nodiscard]
```

Apply an operand's physical transformation to its cached fixed-mesh samples.

8.47.3.56 mask_residual_dual()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::mask_residual_dual (
                const NodalData< EquationT, SchemaT > & data,
                const ::mfem::Vector & dual,
                ::mfem::Vector & volume_dual)
```

Remove constrained rows before applying the volume Jacobian transpose.

8.47.3.57 nodal_work_report()

```
template<typename CacheT>
NumericalWork serif::operators::backend::mfem::nodal_work_report (
                const CacheT & cache)   [nodiscard]
```

Derive numerical work from the dependency report, without a second report lifecycle.

8.47.3.58 prepare_nodal()

```
template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
void serif::operators::backend::mfem::prepare_nodal (
                CacheT & cache,
                const DiscretizationT & discretization,
                const CoefficientsT & term_coefficients,
                const PreparationMode mode = PreparationMode::reuse_unchanged)
```

Translate a numerical rejection at the caller-facing throwing boundary.

8.47.3.59 prepare_point_border()

```
template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
void serif::operators::backend::mfem::prepare_point_border (
                CacheT & cache,
                const DiscretizationT & discretization,
                const CoefficientsT & term_coefficients,
                const PreparationMode mode = PreparationMode::reuse_unchanged)
```

Translate an affine border's numerical rejection at the caller-facing throwing boundary.

8.47.3.60 prepare_volume()

```
template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
void serif::operators::backend::mfem::prepare_volume (
                CacheT & cache,
                const DiscretizationT & discretization,
                const CoefficientsT & term_coefficients,
                const ::mfem::Vector & displacement_coefficients,
                PreparationMode mode = PreparationMode::reuse_unchanged)
```

Throw a caller-visible numerical error when a complete preparation cannot be accepted.

8.47.3.61 report_contribution_preparation()

```
template<VolumeEquation EquationT, std::size_t Index, typename CacheT>
void serif::operators::backend::mfem::report_contribution_preparation (
                const CacheT & cache,
                NumericalWork & report)
```

Convert the dependency engine's node statuses to numerical work counts for one contribution.

8.47.3.62 sample_contribution_basis()

```
template<VolumeEquation EquationT, typename SchemaT, typename ContributionT>
auto serif::operators::backend::mfem::sample_contribution_basis (
                const VolumeConfiguration< EquationT, SchemaT > & configuration,
                const ContributionBinding< ContributionT > & contribution,
                const BasisBindings< EquationPlan< EquationT >::column_count > & bindings,
                TensorBasisCache & cache)   [nodiscard]
```

Sample the reference discretization once for each supported element of this contribution.

8.47.3.63 sample_contribution_operand()

```
template<VolumeEquation EquationT, typename OperandT, typename BindingT>
OperandSamples serif::operators::backend::mfem::sample_contribution_operand (
                const ElementBatch< ElementBasis< EquationPlan< EquationT >::operand_count > > & basis,
                const CoefficientView< BindingT > & input,
                Workspace & workspace)   [nodiscard]
```

Only this operand's coefficients and reference basis determine its reference samples.

8.47.3.64 sample_element_basis()

```
template<VolumeEquation EquationT, typename ContributionT>
ElementBasis< EquationPlan< EquationT >::operand_count > serif::operators::backend::mfem::sample_element_basis (
                const ContributionT & equation,
                int element,
                const discretization::backend::mfem::DofMap & row,
                const discretization::backend::mfem::DofMap & displacement_binding,
                const std::array< const discretization::backend::mfem::DofMap *, EquationPlan< EquationT >::column_count
> & columns,
                const mapping::DomainMapper & mapper,
                const discretization::quadrature::Policy & policy,
                TensorBasisCache & basis_cache,
                ContractionMode contraction_mode)   [nodiscard]
```

Sample the contribution's quadrature rule and bases on the fixed computational mesh.

8.47.3.65 select_nodal_values()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
SelectedNodalValues serif::operators::backend::mfem::select_nodal_values (
                const NodalDiscretization< EquationT, SchemaT > & discretization,
                const CoefficientsT & term_coefficients)   [nodiscard]
```

Validate the whole supplied vector, but cache only coefficients selected by this equation.

8.47.3.66 store_element_linearization()

```
template<VolumeEquation EquationT, typename ContributionT>
bool serif::operators::backend::mfem::store_element_linearization (
                const equations::PointLinearization< typename ContributionT::Operands, typename OperandEvaluator<
equations::TestOperandOf< ContributionT > >::Value > & physics,
                int point,
                double weight,
                ElementData< EquationPlan< EquationT >::operand_count > & data)   [nodiscard]
```

Weight a point's residual and derivatives; reject any non-finite stored component.

8.47.3.67 store_nodal_linearization()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
std::expected< void, PreparationRejection > serif::operators::backend::mfem::store_nodal_linearization (
                const equations::PointLinearization< typename EquationT::Operands, discretization::forms::OperandValue<
equations::TestOperandOf< EquationT > > > & physics,
                const int first,
                NodalData< EquationT, SchemaT > & data)   [nodiscard]
```

Accept only finite, position-independent linearizations before exposing their row data.

8.47.3.68 tensor_element_count()

```
template<VolumeEquation EquationT, typename SchemaT>
std::size_t serif::operators::backend::mfem::tensor_element_count (
                const VolumeData< EquationT, SchemaT > & data)   [nodiscard]
```

Count contribution-elements using tensor contractions for any sampled basis.

8.47.3.69 try_prepare_nodal()

```
template<typename CacheT, NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename
CoefficientsT>
std::expected< void, PreparationRejection > serif::operators::backend::mfem::try_prepare_nodal (
                CacheT & cache,
                const NodalDiscretization< EquationT, SchemaT > & discretization,
                const CoefficientsT & term_coefficients,
                const PreparationMode mode = PreparationMode::reuse_unchanged)   [nodiscard]
```

Validate and gather the selected coefficients, then let the dependency cache decide what needs evaluation.

8.47.3.70 try_prepare_point_border()

```
template<typename CacheT, AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename
CoefficientsT>
std::expected< void, PreparationRejection > serif::operators::backend::mfem::try_prepare_point_border (
                CacheT & cache,
                const PointBorderDiscretization< EquationT, SchemaT > & discretization,
                const CoefficientsT & term_coefficients,
                const PreparationMode mode = PreparationMode::reuse_unchanged)   [nodiscard]
```

Validate both point and scalar sources before publishing the complete border calculation.

8.47.3.71 try_prepare_validated()

```
template<typename CacheT, typename PrepareT>
std::expected< void, PreparationRejection > serif::operators::backend::mfem::try_prepare_validated (
                CacheT & cache,
                PrepareT && prepare)   [nodiscard]
```

A failed validation or calculation must make the previous accepted result inaccessible.

8.47.3.72 try_prepare_volume()

```
template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
std::expected< void, PreparationRejection > serif::operators::backend::mfem::try_prepare_volume (
                CacheT & cache,
                const DiscretizationT & discretization,
                const CoefficientsT & term_coefficients,
                const ::mfem::Vector & displacement_coefficients,
                PreparationMode mode = PreparationMode::reuse_unchanged)   [nodiscard]
```

Validate current inputs and atomically publish a complete volume linearization.

8.47.3.73 validate_nodal_discretization()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::validate_nodal_discretization (
                const NodalDiscretization< EquationT, SchemaT > & discretization)
```

Admit only nodal spaces and the complete selection required by the equation's point or boundary support.

8.47.3.74 validate_nodal_output()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::validate_nodal_output (
                const NodalData< EquationT, SchemaT > & data,
                const ::mfem::Vector & result)
```

A frozen linearization still requires a current selection and a correctly sized output.

8.47.3.75 validate_nodal_selection()

```
template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::validate_nodal_selection (
        const NodalDiscretization< EquationT, SchemaT > & discretization)
```

Check stored selection invariants before dereferencing or gathering any coefficient.

8.47.3.76 validate_point_border_outputs()

```
template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
void serif::operators::backend::mfem::validate_point_border_outputs (
        const PointBorderData< EquationT, SchemaT > & data,
        const ::mfem::Vector & carrier,
        const ::mfem::Vector & phase)
```

Point borders accept one carrier vector and one global scalar output.

8.47.3.77 validate_volume_data()

```
template<VolumeEquation EquationT, typename SchemaT>
void serif::operators::backend::mfem::validate_volume_data (
        const VolumeData< EquationT, SchemaT > & data)
```

Reject changes to discretization objects borrowed by an accepted linearization.

8.47.3.78 validate_volume_discretization()

```
template<typename TestT, typename SchemaT>
void serif::operators::backend::mfem::validate_volume_discretization (
        const VolumeDiscretization< TestT, SchemaT > & discretization)
```

Check borrowed mesh objects before sampling or applying accepted results.

8.47.3.79 volume_height()

```
template<VolumeEquation EquationT, typename SchemaT>
int serif::operators::backend::mfem::volume_height (
        const VolumeData< EquationT, SchemaT > & data)    [nodiscard]
```

Return the supported residual size.

8.47.3.80 volume_residual()

```
template<VolumeEquation EquationT, typename SchemaT>
void serif::operators::backend::mfem::volume_residual (
                const VolumeData< EquationT, SchemaT > & data,
                ::mfem::Vector & result)
```

Copy the residual already assembled and checked when this result was accepted.

8.47.3.81 volume_width()

```
template<VolumeEquation EquationT, typename SchemaT>
int serif::operators::backend::mfem::volume_width (
                const VolumeData< EquationT, SchemaT > & data)  [nodiscard]
```

Packed directions contain unique coefficient terms followed by displacement.

8.47.3.82 volume_work_report()

```
template<typename CacheT>
NumericalWork serif::operators::backend::mfem::volume_work_report (
                const CacheT & cache)  [nodiscard]
```

Derive work counts from accepted node statuses; no second report lifecycle is stored.

8.47.3.83 VolumeDiscretization()

```
template<typename TestT, typename SchemaT>
serif::operators::backend::mfem::VolumeDiscretization (
                const discretization::backend::mfem::TermBinding< TestT, SchemaT > & ,
                const discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, SchemaT > & ,
                const ::mfem::GridFunction & ,
                const mapping::DomainMapper & ) -> VolumeDiscretization< TestT, SchemaT >
```

8.47.4 Variable Documentation

8.47.4.1 index

```
template<VolumeEquation EquationT, EvaluableOperand OperandT, typename BindingT>
std::size_t serif::operators::backend::mfem::index = equations::OperandValues<typename Plan::Operands>::template
index<OperandT>()  [nodiscard], [constexpr]
```

Contract one operand's fixed-mesh basis with its current coefficients.

8.47.4.2 values

return serif::operators::backend::mfem::values

8.48 serif::operators::backend::mfem::nodal_keys Namespace Reference

Classes

- struct BorderLinearization
- struct Linearization

8.49 serif::operators::backend::mfem::volume_keys Namespace Reference

Classes

- struct Basis
- struct Geometry
- struct Linearization
- struct PhysicalValues
- struct QuadratureLinearization
- struct ReferenceValues

8.50 serif::operators::detail Namespace Reference

Classes

- struct AsKeyList
- struct AsKeyList< utils::types::TypeList< T... > >
- struct Columns
- struct Columns< utils::types::TypeList< O... > >
- struct ContributionChildren
- struct ContributionOperands
- struct ContributionOperands< std::tuple< C... > >
- struct CouplingForm
- struct Join

Functions

- template<typename EquationT>
  auto bind_contributions (std::shared_ptr< const EquationT > equation, double coefficient)

  Flatten equation arithmetic while preserving every generated tuple and borrowed parent.
- template<typename EquationT, typename FunctionT>
  void visit_contributions (const EquationT &equation, FunctionT &function, std::size_t &index, double coefficient)

  Visit arithmetic leaves without retaining them beyond the supplied callback.

8.50.1 Function Documentation

8.50.1.1 bind_contributions()

```
template<typename EquationT>
auto serif::operators::detail::bind_contributions (
        std::shared_ptr< const EquationT > equation,
        double coefficient)  [nodiscard]
```

Flatten equation arithmetic while preserving every generated tuple and borrowed parent.

8.50.1.2 visit_contributions()

```
template<typename EquationT, typename FunctionT>
void serif::operators::detail::visit_contributions (
        const EquationT & equation,
        FunctionT & function,
        std::size_t & index,
        double coefficient)
```

Visit arithmetic leaves without retaining them beyond the supplied callback.

8.51 serif::utils Namespace Reference

Namespaces

- namespace checks
- namespace dependencies
- namespace errors
- namespace misc
- namespace types
- namespace user_options
- namespace warnings

Classes

- class CompensatedSum

  Neumaier summation retains small terms when a scalar reduction contains cancellation.

8.52 serif::utils::checks Namespace Reference

Functions

- bool is_finite (const mfem::Vector &vector)
- bool is_finite (const mfem::DenseMatrix &matrix)
- template<typename Scalar>
  requires requires (Scalar x) {{std::isfinite(x)} -> std::convertible_to<bool>; }
  bool is_finite (const Scalar scalar)

8.52.1 Function Documentation

8.52.1.1 is_finite() [1/3]

```
bool serif::utils::checks::is_finite (
            const mfem::DenseMatrix & matrix)   [inline]
```

8.52.1.2 is_finite() [2/3]

```
bool serif::utils::checks::is_finite (
            const mfem::Vector & vector)   [inline]
```

8.52.1.3 is_finite() [3/3]

```
template<typename Scalar>
requires requires (Scalar x) {{std::isfinite(x)} -> std::convertible_to<bool>; }
bool serif::utils::checks::is_finite (
            const Scalar scalar)
```

8.53 serif::utils::dependencies Namespace Reference

Namespaces

- namespace detail
- namespace keys

Classes

- struct Binding
- struct Bindings
- struct BoundType
- class Computation
- class ComputationCache
- class ComputationGraph
- class ComputationStore

  Own independent computation caches under compile-time role keys; this store does not connect their graphs.
- struct ComputationUpdate
- class DependencyContext
- class DependencyContextView
- class DependencyContextView< ParentT, KeyList< Keys... > >
- struct HasBinding
- class IncrementalComputation

  A calculation may reuse unchanged pieces of its previous immutable result.
- struct KeyList

  An ordered collection of role keys, independent of their associated value types.

- struct KeyListBindings
- struct KeyListBindings< BindingsT, KeyList< KeyTs... > >
- class OwnedComputation
- struct PreparationFailure
- struct PreparationReport
- class ProjectedSnapshot
- struct ResolveKeys
- struct ResolveKeys< BindingsT, KeyList< KeyTs... > >
- class Resource
- struct ValueSnapshot

  Default tracking owns an exact value copy; noncopyable resources supply another policy.

Concepts

- concept IsBinding
- concept IsBindings
- concept BindingsWithBinding
- concept IsComputation
- concept CompatibleComputations
- concept IsComputationGraph
- concept IsOwnedComputation
- concept IsDependencyContext
- concept CanPrepare
- concept ComputationGraphFor
- concept IsKeyList
- concept SatisfiesKeyList
- concept IsResource
- concept SnapshotPolicyFor

Typedefs

- template<typename BindingsT, typename KeyT>
  using bound_type_t = BoundType<BindingsT, KeyT>::target_type
- template<typename... KeyTs>
  using InputList = KeyList<KeyTs...>

  Computation arguments are supplied in this key order.
- template<typename... KeyTs>
  using RequestedResults = KeyList<KeyTs...>

  Preparing these results also prepares their required intermediate computations.
- template<typename BindingsT, typename KeysT>
  using ResolvedTargets = ResolveKeys<BindingsT, std::remove_cvref_t<KeysT>>::type
- template<IsKeyList... Lists>
  using CombinedKeyList = detail::MergeKeyLists<KeyList<>, std::remove_cvref_t<Lists>...>::type

  Combine requirements in first-appearance order, retaining distinct keys with shared targets.

Enumerations

- enum class PreparationMode { reuse_unchanged , rebuild }
- enum class PreparationStatus { not_required , reused , computed }

Functions

- template<typename KeyT, IsKeyList InputsT, typename FunctionT>
  requires std::constructible_from<std::decay_t<FunctionT>, FunctionT>
  auto make_computation (FunctionT &&function)

  Attach an output key and ordered input keys to a lambda, function, or other callable.
- template<typename ErrorT = std::string, IsComputationGraph GraphT, IsDependencyContext ContextT>
  requires CanPrepare<GraphT, ContextT, ErrorT> && std::constructible_from<std::remove_cvref_t<GraphT>, GraphT>
  auto make_computation_cache (GraphT &&graph, const ContextT &)

  Fix the graph's source types without retaining the supplied context or evaluating any computations.
- template<typename ErrorT = std::string, IsDependencyContext ContextT, typename... Computations>
  requires CompatibleComputations<Computations...> && (std::constructible_from<std::remove_cvref_t<Computations>, Computations> && ...) && CanPrepare<ComputationGraph<std::remove_cvref_t<Computations>...>, ContextT, ErrorT>
  auto make_computation_cache (const ContextT &context, Computations &&... computations)

  Build a checked graph and an empty cache directly from named computations.
- template<typename... Computations>
  requires CompatibleComputations<Computations...> && (std::constructible_from<std::remove_cvref_t<Computations>, Computations> && ...)
  auto make_computation_graph (Computations &&... computations)

  Reject duplicate producers and cyclic preparation dependencies at the declaration site.
- template<typename KeyT, typename ComputationT>
  requires std::constructible_from<std::remove_cvref_t<ComputationT>, ComputationT>
  auto own_computation (ComputationT &&computation)

  Name an owned computation; rvalues transfer ownership and copyable lvalues are copied.
- template<typename... EntryTs>
  requires (IsOwnedComputation<EntryTs> && ...) && misc::traits::all_unique_types_v<typename std::remove_cvref_t<EntryTs>::Key...> && (std::constructible_from<std::remove_cvref_t<EntryTs>, EntryTs> && ...)
  auto make_computation_store (EntryTs &&... entries)

  Construct the model's computation owner, rejecting duplicate role keys before execution.
- template<typename... Resources>
  requires (IsResource<Resources> && ...) && misc::traits::all_unique_types_v<typename std::remove_cvref_t<Resources>::Key...>
  auto make_dependency_context (Resources &&... resources)

  Build a context whose schema follows directly from its resource keys and types.
- template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
  requires std::is_object_v<ValueT> && std::move_constructible<ValueT> && std::constructible_from<std::decay_t<FunctionT>, FunctionT>
  auto make_incremental_computation (FunctionT &&function)

  Declare the result type explicitly so an incremental callable can inspect its previous value.
- template<typename KeyT, typename ValueT, typename PolicyT = ValueSnapshot>
  auto own (ValueT &&value, PolicyT policy={})

  Store a value in the context; rvalues transfer ownership and lvalues are copied.
- template<typename KeyT, typename ValueT, typename PolicyT = ValueSnapshot>
  auto borrow (ValueT &value, PolicyT policy={})

  Bind an existing object whose lifetime must cover every use of the context.
- template<typename KeyT, typename ValueT, typename PolicyT = ValueSnapshot>
  auto borrow (const ValueT &&, PolicyT={})=delete

  Borrowed storage must never refer to a temporary, including a const temporary.

- template<typename FunctionT>
  auto track_by (FunctionT &&function)

  Select the state that determines whether a source needs recomputation.
- template<typename PolicyT, typename ValueT>
  requires SnapshotPolicyFor<PolicyT, ValueT>
  const void * source_identity (const PolicyT &policy, const ValueT &value)

  Views may identify their underlying resource; ordinary sources retain object identity.

Variables

- template<typename T>
  constexpr bool is_bindings_v = false
- template<IsBinding... BindingTs>
  constexpr bool is_bindings_v< Bindings< BindingTs... > > = misc::traits::all_unique_types_v<typename BindingTs::Key...>
- template<IsBindings BindingsT, typename KeyT>
  constexpr bool has_binding_v = HasBinding<BindingsT, KeyT>::value
- template<typename T>
  constexpr bool is_computation_v = false
- template<typename KeyT, typename InputsT, typename FunctionT>
  constexpr bool is_computation_v< Computation< KeyT, InputsT, FunctionT > > = true
- template<typename T>
  constexpr bool is_computation_graph_v = false
- template<typename... Computations>
  constexpr bool is_computation_graph_v< ComputationGraph< Computations... > > = true
- template<typename T>
  constexpr bool is_owned_computation_v = false
- template<typename KeyT, typename ComputationT>
  constexpr bool is_owned_computation_v< OwnedComputation< KeyT, ComputationT > > = true
- template<typename T>
  constexpr bool is_dependency_context_v = false
- template<typename... Resources>
  constexpr bool is_dependency_context_v< DependencyContext< Resources... > > = true
- template<typename ParentT, typename KeysT>
  constexpr bool is_dependency_context_v< DependencyContextView< ParentT, KeysT > > = true
- template<typename KeyT, typename ValueT, typename InputsT, typename FunctionT>
  constexpr bool is_computation_v< IncrementalComputation< KeyT, ValueT, InputsT, FunctionT > > = true
- template<typename T>
  constexpr bool is_key_list_v = false
- template<typename... KeyTs>
  constexpr bool is_key_list_v< KeyList< KeyTs... > > = misc::traits::all_unique_types_v<KeyTs...>
- std::unordered_map< PreparationStatus, std::string_view > preparation_status_name_map = meta::enum_to_enum_string_map<PreparationStatus>()
- std::unordered_map< PreparationMode, std::string_view > preparation_mode_name_map = meta::enum_to_enum_string_map<PreparationMode>()
- template<typename T>
  constexpr bool is_resource_v = false
- template<typename KeyT, typename StorageT, typename PolicyT>
  constexpr bool is_resource_v< Resource< KeyT, StorageT, PolicyT > > = true

8.53.1 Typedef Documentation

8.53.1.1 bound_type_t

template<typename BindingsT, typename KeyT>
using serif::utils::dependencies::bound_type_t = BoundType<BindingsT, KeyT>::target_type

8.53.1.2 CombinedKeyList

template<IsKeyList... Lists>
using serif::utils::dependencies::CombinedKeyList = detail::MergeKeyLists<KeyList<>, std::remove_cvref_t<Lists>...>::type

Combine requirements in first-appearance order, retaining distinct keys with shared targets.

8.53.1.3 InputList

template<typename... KeyTs>
using serif::utils::dependencies::InputList = KeyList<KeyTs...>

Computation arguments are supplied in this key order.

8.53.1.4 RequestedResults

template<typename... KeyTs>
using serif::utils::dependencies::RequestedResults = KeyList<KeyTs...>

Preparing these results also prepares their required intermediate computations.

8.53.1.5 ResolvedTargets

template<typename BindingsT, typename KeysT>
using serif::utils::dependencies::ResolvedTargets = ResolveKeys<BindingsT, std::remove_cvref_t<KeysT>>::type

8.53.2 Enumeration Type Documentation

8.53.2.1 PreparationMode

enum class serif::utils::dependencies::PreparationMode [strong]

Enumerator

reuse_unchanged	
rebuild	

8.53.2.2 PreparationStatus

enum class serif::utils::dependencies::PreparationStatus [strong]

Enumerator

not_required	
reused	
computed	

8.53.3 Function Documentation

8.53.3.1 borrow() [1/2]

```
template<typename KeyT, typename ValueT, typename PolicyT = ValueSnapshot>
auto serif::utils::dependencies::borrow (
                const ValueT && ,
                PolicyT  = {})   [delete]
```

Borrowed storage must never refer to a temporary, including a const temporary.

8.53.3.2 borrow() [2/2]

```
template<typename KeyT, typename ValueT, typename PolicyT = ValueSnapshot>
auto serif::utils::dependencies::borrow (
                ValueT & value,
                PolicyT policy = {})   [nodiscard]
```

Bind an existing object whose lifetime must cover every use of the context.

8.53.3.3 make_computation()

```
template<typename KeyT, IsKeyList InputsT, typename FunctionT>
requires std::constructible_from<std::decay_t<FunctionT>, FunctionT>
auto serif::utils::dependencies::make_computation (
                FunctionT && function)   [nodiscard]
```

Attach an output key and ordered input keys to a lambda, function, or other callable.

8.53.3.4 make_computation_cache() [1/2]

```
template<typename ErrorT = std::string, IsDependencyContext ContextT, typename... Computations>
requires CompatibleComputations<Computations...> && (std::constructible_from<std::remove_cvref_t<Computations>,
Computations> && ...) && CanPrepare<ComputationGraph<std::remove_cvref_t<Computations>...>, ContextT, ErrorT>
auto serif::utils::dependencies::make_computation_cache (
                const ContextT & context,
                Computations &&... computations)   [nodiscard]
```

Build a checked graph and an empty cache directly from named computations.

8.53.3.5 make_computation_cache() [2/2]

```
template<typename ErrorT = std::string, IsComputationGraph GraphT, IsDependencyContext ContextT>
requires CanPrepare<GraphT, ContextT, ErrorT> && std::constructible_from<std::remove_cvref_t<GraphT>, GraphT>
auto serif::utils::dependencies::make_computation_cache (
                GraphT && graph,
                const ContextT & )  [nodiscard]
```

Fix the graph's source types without retaining the supplied context or evaluating any computations.

8.53.3.6 make_computation_graph()

```
template<typename... Computations>
requires CompatibleComputations<Computations...> && (std::constructible_from<std::remove_cvref_t<Computations>,
Computations> && ...)
auto serif::utils::dependencies::make_computation_graph (
                Computations &&... computations)  [nodiscard]
```

Reject duplicate producers and cyclic preparation dependencies at the declaration site.

8.53.3.7 make_computation_store()

```
template<typename... EntryTs>
requires (IsOwnedComputation<EntryTs> && ...) && misc::traits::all_unique_types_v<typename std::remove_cvref_↩
t<EntryTs>::Key...> && (std::constructible_from<std::remove_cvref_t<EntryTs>, EntryTs> && ...)
auto serif::utils::dependencies::make_computation_store (
                EntryTs &&... entries)  [nodiscard]
```

Construct the model's computation owner, rejecting duplicate role keys before execution.

8.53.3.8 make_dependency_context()

```
template<typename... Resources>
requires (IsResource<Resources> && ...) && misc::traits::all_unique_types_v<typename std::remove_cvref_t<Resources>↩
::Key...>
auto serif::utils::dependencies::make_dependency_context (
                Resources &&... resources)  [nodiscard]
```

Build a context whose schema follows directly from its resource keys and types.

8.53.3.9 make_incremental_computation()

```
template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
requires std::is_object_v<ValueT> && std::move_constructible<ValueT> && std::constructible_from<std::decay_↩
t<FunctionT>, FunctionT>
auto serif::utils::dependencies::make_incremental_computation (
                FunctionT && function)  [nodiscard]
```

Declare the result type explicitly so an incremental callable can inspect its previous value.

8.53.3.10 own()

```
template<typename KeyT, typename ValueT, typename PolicyT = ValueSnapshot>
auto serif::utils::dependencies::own (
                ValueT && value,
                PolicyT policy = {})   [nodiscard]
```

Store a value in the context; rvalues transfer ownership and lvalues are copied.

8.53.3.11 own_computation()

```
template<typename KeyT, typename ComputationT>
requires std::constructible_from<std::remove_cvref_t<ComputationT>, ComputationT>
auto serif::utils::dependencies::own_computation (
                ComputationT && computation)   [nodiscard]
```

Name an owned computation; rvalues transfer ownership and copyable lvalues are copied.

8.53.3.12 source_identity()

```
template<typename PolicyT, typename ValueT>
requires SnapshotPolicyFor<PolicyT, ValueT>
const void * serif::utils::dependencies::source_identity (
                const PolicyT & policy,
                const ValueT & value)   [nodiscard]
```

Views may identify their underlying resource; ordinary sources retain object identity.

8.53.3.13 track_by()

```
template<typename FunctionT>
auto serif::utils::dependencies::track_by (
                FunctionT && function)   [nodiscard]
```

Select the state that determines whether a source needs recomputation.

8.53.4 Variable Documentation

8.53.4.1 has_binding_v

```
template<IsBindings BindingsT, typename KeyT>
bool serif::utils::dependencies::has_binding_v = HasBinding<BindingsT, KeyT>::value   [inline], [constexpr]
```

8.53.4.2 is_bindings_v

```
template<typename T>
bool serif::utils::dependencies::is_bindings_v = false   [inline], [constexpr]
```

8.53.4.3 is_bindings_v< Bindings< BindingTs... > >

template<IsBinding... BindingTs>
bool serif::utils::dependencies::is_bindings_v< Bindings< BindingTs... > > = misc::traits::all_unique_types_v<typename BindingTs::Key...> [inline], [constexpr]

8.53.4.4 is_computation_graph_v

template<typename T>
bool serif::utils::dependencies::is_computation_graph_v = false [inline], [constexpr]

8.53.4.5 is_computation_graph_v< ComputationGraph< Computations... > >

template<typename... Computations>
bool serif::utils::dependencies::is_computation_graph_v< ComputationGraph< Computations... > > = true [inline], [constexpr]

8.53.4.6 is_computation_v

template<typename T>
bool serif::utils::dependencies::is_computation_v = false [inline], [constexpr]

8.53.4.7 is_computation_v< Computation< KeyT, InputsT, FunctionT > >

template<typename KeyT, typename InputsT, typename FunctionT>
bool serif::utils::dependencies::is_computation_v< Computation< KeyT, InputsT, FunctionT > > = true [inline], [constexpr]

8.53.4.8 is_computation_v< IncrementalComputation< KeyT, ValueT, InputsT, FunctionT > >

template<typename KeyT, typename ValueT, typename InputsT, typename FunctionT>
bool serif::utils::dependencies::is_computation_v< IncrementalComputation< KeyT, ValueT, InputsT, FunctionT > > = true [inline], [constexpr]

8.53.4.9 is_dependency_context_v

template<typename T>
bool serif::utils::dependencies::is_dependency_context_v = false [inline], [constexpr]

8.53.4.10 is_dependency_context_v< DependencyContext< Resources... > >

template<typename... Resources>
bool serif::utils::dependencies::is_dependency_context_v< DependencyContext< Resources... > > = true [inline], [constexpr]

8.53.4.11 is_dependency_context_v< DependencyContextView< ParentT, KeysT > >

template<typename ParentT, typename KeysT>
bool serif::utils::dependencies::is_dependency_context_v< DependencyContextView< ParentT, KeysT > > = true [inline], [constexpr]

8.53.4.12 is_key_list_v

template<typename T>
bool serif::utils::dependencies::is_key_list_v = false [inline], [constexpr]

8.53.4.13 is_key_list_v< KeyList< KeyTs... > >

template<typename... KeyTs>
bool serif::utils::dependencies::is_key_list_v< KeyList< KeyTs... > > = misc::traits::all_unique_types_v<KeyTs...> [inline], [constexpr]

8.53.4.14 is_owned_computation_v

template<typename T>
bool serif::utils::dependencies::is_owned_computation_v = false [inline], [constexpr]

8.53.4.15 is_owned_computation_v< OwnedComputation< KeyT, ComputationT > >

template<typename KeyT, typename ComputationT>
bool serif::utils::dependencies::is_owned_computation_v< OwnedComputation< KeyT, ComputationT > > = true [inline], [constexpr]

8.53.4.16 is_resource_v

template<typename T>
bool serif::utils::dependencies::is_resource_v = false [inline], [constexpr]

8.53.4.17 is_resource_v< Resource< KeyT, StorageT, PolicyT > >

template<typename KeyT, typename StorageT, typename PolicyT>
bool serif::utils::dependencies::is_resource_v< Resource< KeyT, StorageT, PolicyT > > = true [inline], [constexpr]

8.53.4.18 preparation_mode_name_map

std::unordered_map<PreparationMode, std::string_view> serif::utils::dependencies::preparation_mode_name_map = meta::enum_to_enum_string_map<PreparationMode>() [inline]

8.53.4.19 preparation_status_name_map

std::unordered_map<PreparationStatus, std::string_view> serif::utils::dependencies::preparation_status_name_map = meta::enum_to_enum_string_map<PreparationStatus>() [inline]

8.54 serif::utils::dependencies::detail Namespace Reference

Classes

- struct AppendBinding
- struct AppendBinding< Bindings< BindingsTs... >, BindingT >
- struct AppendKeys
- struct AppendKeys< ResultT, FirstT, Rest... >
- struct AppendUnique
- struct AppendUnique< KeyList< Keys... >, T >
- struct BorrowedValue
- struct BoundType
- struct BoundType< Bindings< BindingTs... >, KeyT >
- struct CachedResult
- struct ComputationCacheStorage
- struct ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indic
- struct ComputationCacheTraits
- struct ComputationCacheTraits< GraphT, ContextT, ErrorT, true >
- struct ComputationGraphSchema
- struct ComputationGraphSchema< GraphT, ContextT, true >
- struct Graph
- struct HasBinding
- struct HasBinding< Bindings< BindingTs... >, KeyT >
- struct Invocation
- struct Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< decltype(std::declval< const Comp
- struct KeyListSize
- struct KeyListSize< KeyList< Keys... > >
- struct MergeKeyLists
- struct MergeKeyLists< ResultT, KeyList< Keys... >, Rest... >
- struct OwnedValue
- struct ResolveComputation
- struct ResolveComputation< ComputationT, BindingsT, true >
- struct ResolveGraph
- struct ResolveGraph< GraphT, BindingsT, Position, false >
- struct ResolveGraph< GraphT, BindingsT, Position, true >
- struct ResolveStep
- struct ResolveStep< GraphT, BindingsT, Position, ComputationT, true >
- struct ResourceKeys
- struct ResourceKeys< types::TypeList< Resources... > >
- struct Result
- struct Result< std::expected< ValueT, ErrorT > >
- struct SourceSnapshot
- struct UsedResource

Typedefs

- template<typename GraphT, typename ContextT>
  using UsedResources = misc::concepts::FilterT<typename ContextT::Resources, UsedResource<GraphT>↩
  ::template Predicate>

Functions

- template<typename GraphT, typename BindingsT, typename ErrorT, std::size_t... Indices>
  consteval bool compatible_errors (std::index_sequence< Indices... >)

  Value-returning calculations can share a graph with expected results carrying its declared error or an exact union alternative.
- template<typename BindingsT, typename... Keys>
  consteval bool independent_results (KeyList< Keys... >)

  A key has exactly one producer: either the context or a computation.
- template<typename KeyT, typename... Types>
  consteval std::size_t type_index (types::TypeList< Types... >)

  Locate a type without requiring the types themselves to be constructible.

Variables

- template<typename Source, typename Target>
  constexpr bool compatible_error_v = std::same_as<Source, Target>

  Ordinary graph failures retain an exact type; numerical conversions must not erase diagnostics.
- template<typename Source, typename... Alternatives>
  constexpr bool compatible_error_v< Source, std::variant< Alternatives... > >

  Independent modules may explicitly aggregate their unchanged failures in an unambiguous tagged union.
- template<typename KeyT, typename KeysT>
  constexpr bool contains_key_v = false
- template<typename KeyT, typename... Keys>
  constexpr bool contains_key_v< KeyT, KeyList< Keys... > > = (std::is_same_v<KeyT, Keys> || ...)
- template<typename RequiredT, typename AvailableT>
  constexpr bool key_list_subset_v = false
- template<typename... Keys, typename AvailableT>
  constexpr bool key_list_subset_v< KeyList< Keys... >, AvailableT > = (contains_key_v<Keys, AvailableT> && ...)

8.54.1 Typedef Documentation

8.54.1.1 UsedResources

template<typename GraphT, typename ContextT>
using serif::utils::dependencies::detail::UsedResources = misc::concepts::FilterT<typename ContextT::Resources, UsedResource<GraphT>↩
::template Predicate>

8.54.2 Function Documentation

8.54.2.1 compatible_errors()

template<typename GraphT, typename BindingsT, typename ErrorT, std::size_t... Indices>
bool serif::utils::dependencies::detail::compatible_errors (
std::index_sequence< Indices... >) [nodiscard], [consteval]

Value-returning calculations can share a graph with expected results carrying its declared error or an exact union alternative.

8.54.2.2 independent_results()

template<typename BindingsT, typename... Keys>
bool serif::utils::dependencies::detail::independent_results (
KeyList< Keys... >) [nodiscard], [consteval]

A key has exactly one producer: either the context or a computation.

8.54.2.3 type_index()

template<typename KeyT, typename... Types>
std::size_t serif::utils::dependencies::detail::type_index (
types::TypeList< Types... >) [nodiscard], [consteval]

Locate a type without requiring the types themselves to be constructible.

8.54.3 Variable Documentation

8.54.3.1 compatible_error_v

template<typename Source, typename Target>
bool serif::utils::dependencies::detail::compatible_error_v = std::same_as<Source, Target> [inline], [constexpr]

Ordinary graph failures retain an exact type; numerical conversions must not erase diagnostics.

8.54.3.2 compatible_error_v< Source, std::variant< Alternatives... > >

template<typename Source, typename... Alternatives>
bool serif::utils::dependencies::detail::compatible_error_v< Source, std::variant< Alternatives... > > [inline], [constexpr]

Initial value:

```
=
    std::same_as<Source, std::variant<Alternatives...» || ((std::same_as<Source, Alternatives> || ...) &&
     std::constructible_from<std::variant<Alternatives...>, Source>)
```

Independent modules may explicitly aggregate their unchanged failures in an unambiguous tagged union.

8.54.3.3 contains_key_v

template<typename KeyT, typename KeysT>
bool serif::utils::dependencies::detail::contains_key_v = false [inline], [constexpr]

8.54.3.4 contains_key_v< KeyT, KeyList< Keys... > >

template<typename KeyT, typename... Keys>
bool serif::utils::dependencies::detail::contains_key_v< KeyT, KeyList< Keys... > > = (std::is_same_v<KeyT, Keys> || ...) [inline], [constexpr]

8.54.3.5 key_list_subset_v

template<typename RequiredT, typename AvailableT>
bool serif::utils::dependencies::detail::key_list_subset_v = false [inline], [constexpr]

8.54.3.6 key_list_subset_v< KeyList< Keys... >, AvailableT >

template<typename... Keys, typename AvailableT>
bool serif::utils::dependencies::detail::key_list_subset_v< KeyList< Keys... >, AvailableT > = (contains_key_v<Keys, AvailableT> && ...) [inline], [constexpr]

8.55 serif::utils::dependencies::keys Namespace Reference

Classes

- struct Coordinates

  Positions supplied to a computation, in the caller's coordinate convention.
- struct Input

  The main input to a computation.
- struct Output

  The main result of a computation.
- struct Parameters

  Configuration values that can affect the result.
- struct Scale

  A multiplicative scale supplied as an input.

8.56 serif::utils::errors Namespace Reference

Classes

- struct serif_error_printer

Enumerations

- enum class SERiFErrorCode : uint16_t {
  UNKNOWN_ERROR = 0 , INVALID_ARGUMENT = 100 , INVALID_QUADRATURE_ORDER
  , INVALID_DOMAIN_ERROR =110 ,
  NEGATIVE_VALUE_ERROR , NON_FINITE_VALUE_ERROR , JACOBIAN_ERROR = 200 ,
  INVERTED_JACOBIAN ,
  NON_FINITE_DETERMINATE , EOS_ERROR = 300 , EOS_UNSUPPORTED_RELATION ,
  EOS_UNSUPPORTED_DERIVATIVE ,
  EOS_WRONG_INPUT_COUNT , EOS_WRONG_INPUT_QUANTITY , EOS_NONFINITE_INPUT
  , EOS_OUTSIDE_DOMAIN ,
  EOS_NONFINITE_RESULT , EOS_INVALID_THERMODYNAMIC_INPUT , INVALID_DIMENSION
  = 400 , NON_FINITE_MAPPING_INPUT ,
  INVALID_REFERENCE_RADIUS , AT_COMPACTIFIED_INFINITY , OUTSIDE_REFERENCE_DOMAIN
  , NON_FINITE_MAPPING_RESULT ,
  NON_POSITIVE_DETERMINATE , INVALID_MAPPING , MISC_MAPPING_ERROR =410 ,
  INVALID_DOF_COUNT ,
  INVALID_ELEMENT_GEOMETRY , INVALID_DOF_ORDERING , INVALID_COMPACTIFICATION_RANK
  , INVALID_COMPACTIFICATION_TYPE ,
  INVALID_COMPACTIFICATION_DERIVATIVE_TYPE , INVALID_VACUUM_ELEMENT_ATTRIBUTE_ID
  , MISSING_EXTERIOR_MAP , COMPACTIFICATION_MISMATCH ,
  MISSING_FACE_ELEMENT , JACOBIAN_SHAPE_ERROR , MISSING_DISPLACEMENT_FINITE_ELEMENT
  , MISMATCHED_MESHING_MAPS ,
  IMPROPERLY_ORDERED_REFERENCE_RADII = 450 , INVALID_COORDINATE_TOLERANCE
  , EVALUATOR_ERROR =460 , CACHE_FE_BINDING_ERROR ,
  INVALID_ELEMENT_ID_ERROR , ELEMENT_NULLPTR_ERROR , MATRIX_SHAPE_ERROR
  = 500 , OUT_OF_RANGE_ERROR ,
  FAILED_TYPE_INVARIANT = 1000 }

Functions

- template<misc::concepts::IsException ExceptionT, SERiFErrorCode code>
  void serif_error (const std::string_view message, auto... args)
- template<SERiFErrorCode code, misc::concepts::IsException ExceptionT = std::logic_error>
  void serif_assert (const bool condition, const std::string_view message, auto... args)
- template<SERiFErrorCode code, types::ct_error_string message, bool condition>
  constexpr void serif_assert ()
- template<SERiFErrorCode code, misc::concepts::IsException ExceptionT = std::logic_error>
  void serif_ensure (const bool condition, const std::string_view message, auto... args)
- template<SERiFErrorCode code, types::ct_error_string message, bool condition>
  constexpr void serif_ensure ()

Variables

- auto SERiFErrorCodeNameToTypeMap = meta::enum_to_string_enum_map<SERiFErrorCode>()
- auto SERiFErrorCodeTypeToNameMap = meta::enum_to_enum_string_map<SERiFErrorCode>()

8.56.1 Enumeration Type Documentation

8.56.1.1 SERiFErrorCode

enum class serif::utils::errors::SERiFErrorCode : uint16_t [strong]

Enumerator

UNKNOWN_ERROR	
INVALID_ARGUMENT	
INVALID_QUADRATURE_ORDER	
INVALID_DOMAIN_ERROR	
NEGATIVE_VALUE_ERROR	
NON_FINITE_VALUE_ERROR	
JACOBIAN_ERROR	
INVERTED_JACOBIAN	
NON_FINITE_DETERMINATE	
EOS_ERROR	
EOS_UNSUPPORTED_RELATION	
EOS_UNSUPPORTED_DERIVATIVE	
EOS_WRONG_INPUT_COUNT	
EOS_WRONG_INPUT_QUANTITY	
EOS_NONFINITE_INPUT	
EOS_OUTSIDE_DOMAIN	
EOS_NONFINITE_RESULT	
EOS_INVALID_THERMODYNAMIC_INPUT	
INVALID_DIMENSION	
NON_FINITE_MAPPING_INPUT	
INVALID_REFERENCE_RADIUS	
AT_COMPACTIFIED_INFINITY	
OUTSIDE_REFERENCE_DOMAIN	
NON_FINITE_MAPPING_RESULT	
NON_POSITIVE_DETERMINATE	
INVALID_MAPPING	
MISC_MAPPING_ERROR	
INVALID_DOF_COUNT	
INVALID_ELEMENT_GEOMETRY	
INVALID_DOF_ORDERING	
INVALID_COMPACTIFICATION_RANK	
INVALID_COMPACTIFICATION_TYPE	
INVALID_COMPACTIFICATION_DERIVATIVE_TYPE	
INVALID_VACUUM_ELEMENT_ATTRIBUTE_ID	
MISSING_EXTERIOR_MAP	
COMPACTIFICATION_MISMATCH	
MISSING_FACE_ELEMENT	
JACOBIAN_SHAPE_ERROR	
MISSING_DISPLACEMENT_FINITE_ELEMENT_SPACE	
MISMATCHED_MESHING_MAPS	
IMPROPERLY_ORDERED_REFERENCE_RADII	
INVALID_COORDINATE_TOLERANCE	
EVALUATOR_ERROR	

Enumerator

CACHE_FE_BINDING_ERROR	
INVALID_ELEMENT_ID_ERROR	
ELEMENT_NULLPTR_ERROR	
MATRIX_SHAPE_ERROR	
OUT_OF_RANGE_ERROR	
FAILED_TYPE_INVARIANT	

8.56.2 Function Documentation

8.56.2.1 serif_assert() [1/2]

template<SERiFErrorCode code, types::ct_error_string message, bool condition>
void serif::utils::errors::serif_assert () [constexpr]

8.56.2.2 serif_assert() [2/2]

template<SERiFErrorCode code, misc::concepts::IsException ExceptionT = std::logic_error>
void serif::utils::errors::serif_assert (
const bool condition,
const std::string_view message,
auto... args)

8.56.2.3 serif_ensure() [1/2]

template<SERiFErrorCode code, types::ct_error_string message, bool condition>
void serif::utils::errors::serif_ensure () [constexpr]

8.56.2.4 serif_ensure() [2/2]

template<SERiFErrorCode code, misc::concepts::IsException ExceptionT = std::logic_error>
void serif::utils::errors::serif_ensure (
const bool condition,
const std::string_view message,
auto... args)

8.56.2.5 serif_error()

template<misc::concepts::IsException ExceptionT, SERiFErrorCode code>
void serif::utils::errors::serif_error (
const std::string_view message,
auto... args)

8.56.3 Variable Documentation

8.56.3.1 SERiFErrorCodeNameToTypeMap

auto serif::utils::errors::SERiFErrorCodeNameToTypeMap = meta::enum_to_string_enum_map<SERiFErrorCode>() [inline]

8.56.3.2 SERiFErrorCodeTypeToNameMap

auto serif::utils::errors::SERiFErrorCodeTypeToNameMap = meta::enum_to_enum_string_map<SERiFErrorCode>() [inline]

8.57 serif::utils::misc Namespace Reference

Namespaces

- namespace concepts
- namespace format
- namespace std_helper
- namespace terminal
- namespace traits

Functions

- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsNumeric Scalar>
  void validate_finite (const Scalar value)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void validate_greater_than (const Scalar value, const Scalar predicate)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void validate_less_than (const Scalar value, const Scalar predicate)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void validate_nonnegativity (const Scalar value)

8.57.1 Function Documentation

8.57.1.1 validate_finite()

template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsNumeric Scalar>
void serif::utils::misc::validate_finite (
const Scalar value)

8.57.1.2 validate_greater_than()

template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
void serif::utils::misc::validate_greater_than (
const Scalar value,
const Scalar predicate)

8.57.1.3 validate_less_than()

template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
void serif::utils::misc::validate_less_than (
const Scalar value,
const Scalar predicate)

8.57.1.4 validate_nonnegativity()

template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
void serif::utils::misc::validate_nonnegativity (
const Scalar value)

8.58 serif::utils::misc::concepts Namespace Reference

Classes

- struct Concat
- struct Concat< types::TypeList< LhsTs... >, types::TypeList< RhsTs... >, Rest... >
- struct Concat< types::TypeList< Ts... > >
- struct Concat<>
- struct EnumerateTypes
- struct EnumerateTypes< types::TypeList< TypeTs... >, Descriptor >
- struct Filter
- struct Filter< types::TypeList< Head, Tail... >, Predicate >
- struct Filter< types::TypeList<>, Predicate >
- struct Prepend
- struct Prepend< T, types::TypeList< Ts... > >

Concepts

- concept IsException
- concept IsExceptionWithErrorCode
- concept IsNumeric
- concept IsOrderable
- concept IsComparableToNumeric
- concept SetOfOneOrTwo
- concept IsEnum
- concept AllTypesAreUnique

Typedefs

- template<typename... Lists>
  using ConcatT = Concat<Lists...>::Type
- template<typename TypeList, template< typename > typename Descriptor>
  using EnumerateTypesT = EnumerateTypes<TypeList, Descriptor>::Type
- template<typename T, typename List>
  using PrependT = Prepend<T, List>::Type
- template<typename List, template< typename > typename Predicate>
  using FilterT = Filter<List, Predicate>::Type

8.58.1 Typedef Documentation

8.58.1.1 ConcatT

template<typename... Lists>
using serif::utils::misc::concepts::ConcatT = Concat<Lists...>::Type

8.58.1.2 EnumerateTypesT

template<typename TypeList, template< typename > typename Descriptor>
using serif::utils::misc::concepts::EnumerateTypesT = EnumerateTypes<TypeList, Descriptor>::Type

8.58.1.3 FilterT

template<typename List, template< typename > typename Predicate>
using serif::utils::misc::concepts::FilterT = Filter<List, Predicate>::Type

8.58.1.4 PrependT

template<typename T, typename List>
using serif::utils::misc::concepts::PrependT = Prepend<T, List>::Type

8.59 serif::utils::misc::format Namespace Reference

Functions

- template<typename T>
  std::string join (const std::span< T > &elements, std::string_view delimiter)

8.59.1 Function Documentation

8.59.1.1 join()

```
template<typename T>
std::string serif::utils::misc::format::join (
                const std::span< T > & elements,
                std::string_view delimiter)
```

8.60 serif::utils::misc::std_helper Namespace Reference

Typedefs

- template<class T>
  using NeitherConstNorVolatile = std::remove_cv_t<T>
- template<class T>
  using NeitherConstNorVolatileReference = std::remove_cvref_t<T>

Functions

- template<typename... Ts>
  consteval std::size_t CountVariadicArguments () noexcept

Variables

- template<class Child, class Parent>
  constexpr bool AIsBaseClassOfB = std::is_base_of_v<Parent, Child>
- template<class Ta, class Tb>
  constexpr bool SameType = std::is_same_v<Ta, Tb>

8.60.1 Typedef Documentation

8.60.1.1 NeitherConstNorVolatile

```
template<class T>
using serif::utils::misc::std_helper::NeitherConstNorVolatile = std::remove_cv_t<T>
```

8.60.1.2 NeitherConstNorVolatileReference

```
template<class T>
using serif::utils::misc::std_helper::NeitherConstNorVolatileReference = std::remove_cvref_t<T>
```

8.60.2 Function Documentation

8.60.2.1 CountVariadicArguments()

template<typename... Ts>
std::size_t serif::utils::misc::std_helper::CountVariadicArguments () [nodiscard], [consteval], [noexcept]

8.60.3 Variable Documentation

8.60.3.1 AIsBaseClassOfB

template<class Child, class Parent>
bool serif::utils::misc::std_helper::AIsBaseClassOfB = std::is_base_of_v<Parent, Child> [constexpr]

8.60.3.2 SameType

template<class Ta, class Tb>
bool serif::utils::misc::std_helper::SameType = std::is_same_v<Ta, Tb> [constexpr]

8.61 serif::utils::misc::terminal Namespace Reference

Enumerations

- enum class ANSIColors : int {
  BLACK = 30 , RED = 31 , GREEN = 32 , YELLOW = 33 ,
  BLUE = 34 , MAGENTA = 35 , CYAN = 36 , WHITE = 37 ,
  BRIGHT_BLACK = 90 , BRIGHT_RED = 91 , BRIGHT_GREEN = 92 , RESET = 0 }

Functions

- consteval std::string_view resolve_ansi_color (const ANSIColors color) noexcept
- bool supports_ansi (std::FILE *stream=stdout) noexcept

Variables

- constexpr std::string_view a_black = resolve_ansi_color(ANSIColors::BLACK)
- constexpr std::string_view a_red = resolve_ansi_color(ANSIColors::RED)
- constexpr std::string_view a_green = resolve_ansi_color(ANSIColors::GREEN)
- constexpr std::string_view a_yellow = resolve_ansi_color(ANSIColors::YELLOW)
- constexpr std::string_view a_blue = resolve_ansi_color(ANSIColors::BLUE)
- constexpr std::string_view a_magenta = resolve_ansi_color(ANSIColors::MAGENTA)
- constexpr std::string_view a_cyan = resolve_ansi_color(ANSIColors::CYAN)
- constexpr std::string_view a_white = resolve_ansi_color(ANSIColors::WHITE)
- constexpr std::string_view a_bright_black = resolve_ansi_color(ANSIColors::BRIGHT_BLACK)
- constexpr std::string_view a_bright_red = resolve_ansi_color(ANSIColors::BRIGHT_RED)
- constexpr std::string_view a_bright_green = resolve_ansi_color(ANSIColors::BRIGHT_GREEN)
- constexpr std::string_view a_reset = resolve_ansi_color(ANSIColors::RESET)

8.61.1 Enumeration Type Documentation

8.61.1.1 ANSIColors

enum class serif::utils::misc::terminal::ANSIColors : int [strong]

Enumerator

BLACK	
RED	
GREEN	
YELLOW	
BLUE	
MAGENTA	
CYAN	
WHITE	
BRIGHT_BLACK	
BRIGHT_RED	
BRIGHT_GREEN	
RESET	

8.61.2 Function Documentation

8.61.2.1 resolve_ansi_color()

std::string_view serif::utils::misc::terminal::resolve_ansi_color (
const ANSIColors color) [consteval], [noexcept]

8.61.2.2 supports_ansi()

bool serif::utils::misc::terminal::supports_ansi (
std::FILE * stream = stdout) [inline], [noexcept]

8.61.3 Variable Documentation

8.61.3.1 a_black

std::string_view serif::utils::misc::terminal::a_black = resolve_ansi_color(ANSIColors::BLACK) [constexpr]

8.61.3.2 a_blue

std::string_view serif::utils::misc::terminal::a_blue = resolve_ansi_color(ANSIColors::BLUE) [constexpr]

8.61.3.3 a_bright_black

std::string_view serif::utils::misc::terminal::a_bright_black = resolve_ansi_color(ANSIColors::BRIGHT_BLACK) [constexpr]

8.61.3.4 a_bright_green

std::string_view serif::utils::misc::terminal::a_bright_green = resolve_ansi_color(ANSIColors::BRIGHT_GREEN) [constexpr]

8.61.3.5 a_bright_red

std::string_view serif::utils::misc::terminal::a_bright_red = resolve_ansi_color(ANSIColors::BRIGHT_RED) [constexpr]

8.61.3.6 a_cyan

std::string_view serif::utils::misc::terminal::a_cyan = resolve_ansi_color(ANSIColors::CYAN) [constexpr]

8.61.3.7 a_green

std::string_view serif::utils::misc::terminal::a_green = resolve_ansi_color(ANSIColors::GREEN) [constexpr]

8.61.3.8 a_magenta

std::string_view serif::utils::misc::terminal::a_magenta = resolve_ansi_color(ANSIColors::MAGENTA) [constexpr]

8.61.3.9 a_red

std::string_view serif::utils::misc::terminal::a_red = resolve_ansi_color(ANSIColors::RED) [constexpr]

8.61.3.10 a_reset

std::string_view serif::utils::misc::terminal::a_reset = resolve_ansi_color(ANSIColors::RESET) [constexpr]

8.61.3.11 a_white

std::string_view serif::utils::misc::terminal::a_white = resolve_ansi_color(ANSIColors::WHITE) [constexpr]

8.61.3.12 a_yellow

std::string_view serif::utils::misc::terminal::a_yellow = resolve_ansi_color(ANSIColors::YELLOW) [constexpr]

8.62 serif::utils::misc::traits Namespace Reference

Variables

- template<typename... Ts>
  constexpr bool all_unique_types_v = true
- template<typename First, typename... Rest>
  constexpr bool all_unique_types_v< First, Rest... > = (!std_helper::SameType<First, Rest> && ...) && all_unique_types_v<Rest...>

8.62.1 Variable Documentation

8.62.1.1 all_unique_types_v

template<typename... Ts>
bool serif::utils::misc::traits::all_unique_types_v = true [constexpr]

8.62.1.2 all_unique_types_v< First, Rest... >

template<typename First, typename... Rest>
bool serif::utils::misc::traits::all_unique_types_v< First, Rest... > = (!std_helper::SameType<First, Rest> && ...) && all_unique_types_v<Rest...> [constexpr]

8.63 serif::utils::types Namespace Reference

Namespaces

- namespace backend

Classes

- struct ct_error_string
- struct ctstring
- struct DenseMatrixBase
- struct DenseTensorBase
- struct SparseMatrixBase
- struct SparseTensorBase
- struct TypeList
- struct VectorBase

8.64 serif::utils::types::backend Namespace Reference

Namespaces

- namespace mfem

8.65 serif::utils::types::backend::mfem Namespace Reference

8.66 serif::utils::user_options Namespace Reference

Classes

- struct DomainMapperOptions
- struct KelvinCompactificationOptions

8.67 serif::utils::warnings Namespace Reference

Functions

- void serif_warning (const std::string_view message, auto... args)

8.67.1 Function Documentation

8.67.1.1 serif_warning()

```
void serif::utils::warnings::serif_warning (
                const std::string_view message,
                auto... args)
```

Chapter 9

Concept Documentation

9.1 @120027171211173332346346153147002002266065223266::HasBoundType Concept Reference

9.1.1 Concept definition

```
template<typename BindingsT, typename KeyT>
concept @120027171211173332346346153147002002266065223266::HasBoundType =  requires {
      typename dependencies::bound_type_t<BindingsT, KeyT>;
   }
```

9.2 @120027171211173332346346153147002002266065223266::HasBoundTypeMember Concept Reference

9.2.1 Concept definition

```
template<typename BindingsT, typename KeyT>
concept @120027171211173332346346153147002002266065223266::HasBoundTypeMember =  requires {
      typename dependencies::BoundType<BindingsT, KeyT>::target_type;
   }
```

9.3 @200074317303167056063064256070046203054064052214::CanInspectChange Concept Reference

9.3.1 Concept definition

```
template<typename KeyT>
concept @200074317303167056063064256070046203054064052214::CanInspectChange =  requires(const Update& update) {
       update.template changed<KeyT>(); }
```

9.4 @20007431730316705606306425607004620305406405221 4::CanMakeCache Concept Reference

9.4.1 Concept definition

```
template<typename ContextT, typename... Computations>
concept @20007431730316705606306425607004620305406405221 4::CanMakeCache =  requires(const ContextT& context,
        Computations&&... computations) {
      dependencies::make_computation_cache(context, std::forward<Computations>(computations)...);
   }
```

9.5 @20007431730316705606306425607004620305406405221 4::CanMakeComputation Concept Reference

9.5.1 Concept definition

```
template<typename FunctionT, typename InputsT dependencies::InputList<Source»
concept @20007431730316705606306425607004620305406405221 4::CanMakeComputation =  requires(FunctionT&& function) {
      dependencies::make_computation<First, InputsT>(std::forward<FunctionT>(function));
   }
```

9.6 @20007431730316705606306425607004620305406405221 4::CanMakeFallibleCache Concept Reference

9.6.1 Concept definition

```
template<typename ErrorT, typename... Computations>
concept @20007431730316705606306425607004620305406405221 4::CanMakeFallibleCache =  requires(const DependencyContext&
       context, Computations&&... computations) {
      dependencies::make_computation_cache<ErrorT>(context, std::forward<Computations>(computations)...);
   }
```

9.7 @20007431730316705606306425607004620305406405221 4::CanRequest Concept Reference

9.7.1 Concept definition

```
template<typename RequestedT>
concept @20007431730316705606306425607004620305406405221 4::CanRequest =  requires(ComputationCache& cache, const
       DependencyContext& context, RequestedT request) {
      cache.try_prepare(context, request);
   }
```

9.8 @23037207512511711214621602205025733117713710732 7::CanCompose Concept Reference

Equal vector lengths cannot substitute for matching physical meanings at a composition boundary.

9.8.1 Concept definition

```
template<typename Surface, typename Extension>
concept @230372075125117112146216022050257331177137107327::CanCompose = requires(std::shared_ptr<const Surface> surface,
        std::shared_ptr<const Extension> extension) {
      deformation::compose_deformation(surface, extension);
   }
```

9.8.2 Detailed Description

Equal vector lengths cannot substitute for matching physical meanings at a composition boundary.

9.9 @247006277077211216153350341142357337161001026143::HasSurface↩ Relation Concept Reference

9.9.1 Concept definition

```
template<typename EOS, typename Bindings>
concept @247006277077211216153350341142357337161001026143::HasSurfaceRelation = requires {
      typename surface::pressure_relation_t<EOS, Bindings>;
   }
```

9.10 @247006277077211216153350341142357337161001026143::Selects↩ Carrier Concept Reference

9.10.1 Concept definition

```
template<typename Candidate>
concept @247006277077211216153350341142357337161001026143::SelectsCarrier = requires {
      typename Candidate::CarrierQuantity;
   }
```

9.11 @252166143035264314021214246026055041120163322041::CanResolve Concept Reference

9.11.1 Concept definition

```
template<typename BindingsT, typename KeysT>
concept @252166143035264314021214246026055041120163322041::CanResolve = requires {
      typename dependencies::ResolvedTargets<BindingsT, KeysT>;
   }
```

9.12 @323172327364102235210333337057065015152354331343::CanBorrow Concept Reference

9.12.1 Concept definition

```
template<typename ValueT>
concept @323172327364102235210333337057065015152354331343::CanBorrow = requires(ValueT&& value) {
      dependencies::borrow<First>(std::forward<ValueT>(value)); }
```

9.13 @32317232736410223521033333705706501515235433134 3::CanMakeContext Concept Reference

9.13.1 Concept definition

```
template<typename... Resources>
concept @32317232736410223521033333705706501515235433134 3::CanMakeContext = requires(Resources&&... resources) {
    dependencies::make_dependency_context(std::forward<Resources>(resources)...);
  }
```

9.14 @32317232736410223521033333705706501515235433134 3::CanRead Concept Reference

9.14.1 Concept definition

```
template<typename ContextT, typename KeyT>
concept @32317232736410223521033333705706501515235433134 3::CanRead = requires(const ContextT& context) { context.template
    get<KeyT>(); }
```

9.15 @32317232736410223521033333705706501515235433134 3::CanSelect Concept Reference

9.15.1 Concept definition

```
template<typename ContextT>
concept @32317232736410223521033333705706501515235433134 3::CanSelect = requires(ContextT&& context) {
    std::forward<ContextT>(context).select(dependencies::KeyList<First>{});
  }
```

9.16 @34714524103410214107200435717127436023034606435 6::CanMakeStore Concept Reference

9.16.1 Concept definition

```
template<typename... EntryTs>
concept @34714524103410214107200435717127436023034606435 6::CanMakeStore = requires(EntryTs&&... entries) {
    dependencies::make_computation_store(std::forward<EntryTs>(entries)...);
  }
```

9.17 @34714524103410214107200435717127436023034606435 6::CanOwn Concept Reference

9.17.1 Concept definition

```
template<typename ValueT>
concept @34714524103410214107200435717127436023034606435 6::CanOwn = requires(ValueT&& value) {
    dependencies::own_computation<First>(std::forward<ValueT>(value));
  }
```

9.18 @3471452410341021410720043571712743602303460643 56::CanRead Concept Reference

9.18.1 Concept definition

```
template<typename StoreT, typename KeyT>
concept @3471452410341021410720043571712743602303460643 56::CanRead =  requires(StoreT&& store) {
        std::forward<StoreT>(store).template get<KeyT>();
    }
```

9.19 @3471452410341021410720043571712743602303460643 56::CanRead↩ Entry Concept Reference

9.19.1 Concept definition

```
template<typename EntryT>
concept @3471452410341021410720043571712743602303460643 56::CanReadEntry =  requires(EntryT&& entry) {
        std::forward<EntryT>(entry).get();
    }
```

9.20 serif::dimensions::PhysicalQuantityType Concept Reference

Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity. Note all ThermodynamicQuantities will also satisfy this concept.

#include <quantities.hpp>

9.20.1 Concept definition

```
template<typename QuantityCandidate>
concept serif::dimensions::PhysicalQuantityType =
        std::same_as<QuantityCandidate, NeitherConstNorVolatile<QuantityCandidate» &&
        std::derived_from<QuantityCandidate, PhysicalQuantity>
```

9.20.2 Detailed Description

Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity. Note all ThermodynamicQuantities will also satisfy this concept.

Template Parameters

QuantityCandidate	The type that will be validated

9.21 serif::dimensions::QuantityValueType Concept Reference

Turn those compile time structs into a concept that can be used to constrain a template parameter.

#include <quantities.hpp>

9.21.1 Concept definition

```
template<typename Candidate>
concept serif::dimensions::QuantityValueType =  IsQuantityValue<std::remove_cvref_t<Candidate>>::value
```

9.21.2 Detailed Description

Turn those compile time structs into a concept that can be used to constrain a template parameter.

Template Parameters

Candidate	Some type that might be a quantity value type

9.22 serif::dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity Concept Reference

#include <concepts.hpp>

9.22.1 Concept definition

```
template<typename Quantity>
concept serif::dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity =
      ThermodynamicQuantityType<Quantity> &&
      requires {{Quantity::identifier} -> std::convertible_to<std::string_view>; } &&
      (std::string_view{Quantity::identifier}.size() > 0)
```

9.22.2 Detailed Description

@breif Concept to limit template arguments such that they are compatible with the expected ThermodynamicQuantity interface. Specifically, a valid type must contain a member identifier which can be converted to std::string_view and that member, when converted to a string_view must be longer than 0 length.

9.23 serif::dimensions::ThermodynamicQuantityType Concept Reference

Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity (both enforced through checking PhysicalQuantityType) and then that is is also derived from ThermodynamicQuantity. Note not all physical quanties will satisfy this.

#include <quantities.hpp>

9.23.1 Concept definition

```
template<typename QuantityCandidate>
concept serif::dimensions::ThermodynamicQuantityType =
      PhysicalQuantityType<QuantityCandidate> &&
      std::derived_from<QuantityCandidate, ThermodynamicQuantity>
```

9.23.2 Detailed Description

Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity (both enforced through checking PhysicalQuantityType) and then that is is also derived from ThermodynamicQuantity. Note not all physical quanties will satisfy this.

Template Parameters

QuantityCandidate	

9.24 serif::dimensions::ThermodynamicQuantityValueType Concept Reference

#include <quantities.hpp>

9.24.1 Concept definition

```
template<typename Candidate>
concept serif::dimensions::ThermodynamicQuantityValueType =  QuantityValueType<Candidate> &&
      ThermodynamicQuantityType<QuantityOfT<Candidate>>
```

9.25 serif::discretization::blocks::Field Concept Reference

Concept for field types in the discretization framework. To be a field a type must derive from the field base class.

#include <concepts.hpp>

9.25.1 Concept definition

```
template<typename FieldCandidate>
concept serif::discretization::blocks::Field =  std::derived_from<FieldCandidate, field>
```

9.25.2 Detailed Description

Concept for field types in the discretization framework. To be a field a type must derive from the field base class.

9.26 serif::discretization::blocks::GlobalScalarTerm Concept Reference

Concept for term types that are global scalar terms. A global scalar term is a term that has support (HasSupport) but does not have a functional space (HasFunctionalSpace) and is derived from the global_↩ scalar class.

#include <concepts.hpp>

9.26.1 Concept definition

```
template<typename T>
concept serif::discretization::blocks::GlobalScalarTerm =  Term<T> && HasSupport<T> && !HasFunctionalSpace<T> &&
        std::derived_from<std::remove_cvref_t<T>, global_scalar>
```

9.26.2 Detailed Description

Concept for term types that are global scalar terms. A global scalar term is a term that has support (HasSupport) but does not have a functional space (HasFunctionalSpace) and is derived from the global_↩ scalar class.

9.27 serif::discretization::blocks::HasFunctionalSpace Concept Reference

Concept for term types that have a functional space. To have a functional space a type must first be a valid Term and must have a FunctionalSpace member type.

#include <concepts.hpp>

9.27.1 Concept definition

```
template<typename T>
concept serif::discretization::blocks::HasFunctionalSpace =
        Term<T> &&
        requires {
        typename std::remove_cvref_t<T>::FunctionalSpace;
        }
```

9.27.2 Detailed Description

Concept for term types that have a functional space. To have a functional space a type must first be a valid Term and must have a FunctionalSpace member type.

9.28 serif::discretization::blocks::HasQuantity Concept Reference

Concept for term types that have a physical quantity associated with them. To have a quantity a type must first be a valid Term and must have a Quantity member type.

#include <concepts.hpp>

9.28.1 Concept definition

```
template<typename T>
concept serif::discretization::blocks::HasQuantity =
        Term<T> &&
        requires {
        typename std::remove_cvref_t<T>::Quantity;
        }
```

9.28.2 Detailed Description

Concept for term types that have a physical quantity associated with them. To have a quantity a type must first be a valid Term and must have a Quantity member type.

9.29 serif::discretization::blocks::HasResidualBlock Concept Reference

Concept for term types that have a residual block associated with them. To have a residual block a type must first be a valid Term and must have a residual member type.

#include <concepts.hpp>

9.29.1 Concept definition

```
template<typename T>
concept serif::discretization::blocks::HasResidualBlock =
        Term<T> &&
        requires {
        typename std::remove_cvref_t<T>::residual;
    }
```

9.29.2 Detailed Description

Concept for term types that have a residual block associated with them. To have a residual block a type must first be a valid Term and must have a residual member type.

9.30 serif::discretization::blocks::HasSupport Concept Reference

Concept for term types that have a domain over which they are supported. To have support a type must first be a valid Term and must have a Support member type.

#include <concepts.hpp>

9.30.1 Concept definition

```
template<typename T>
concept serif::discretization::blocks::HasSupport =
        Term<T> &&
        requires {
        typename std::remove_cvref_t<T>::Support;
        }
```

9.30.2 Detailed Description

Concept for term types that have a domain over which they are supported. To have support a type must first be a valid Term and must have a Support member type.

9.31 serif::discretization::blocks::HasValueBlock Concept Reference

Concept for term types that have a value block associated with them. To have a value block a type must first be a valid Term and must have a value member type.

#include <concepts.hpp>

9.31.1 Concept definition

```
template<typename T>
concept serif::discretization::blocks::HasValueBlock =
    Term<T> &&
    requires {
    typename std::remove_cvref_t<T>::value;
    }
```

9.31.2 Detailed Description

Concept for term types that have a value block associated with them. To have a value block a type must first be a valid Term and must have a value member type.

9.32 serif::discretization::blocks::PhysicalSpatialTerm Concept Reference

Concept for a physical spatial term. A physical spatial term is a spatial term (SpatialTerm) that has a physical quantity associated with it (HasQuantity).

#include <concepts.hpp>

9.32.1 Concept definition

```
template<typename T>
concept serif::discretization::blocks::PhysicalSpatialTerm =
    SpatialTerm<T> &&
    HasQuantity<T>
```

9.32.2 Detailed Description

Concept for a physical spatial term. A physical spatial term is a spatial term (SpatialTerm) that has a physical quantity associated with it (HasQuantity).

9.33 serif::discretization::blocks::SpatialTerm Concept Reference

Concept for term types that are spatial terms. A spatial term is a term that has a functional space (HasFunctionalSpace) and a support (HasSupport).

#include <concepts.hpp>

9.33.1 Concept definition

```
template<typename T>
concept serif::discretization::blocks::SpatialTerm =
    Term<T> &&
    HasFunctionalSpace<T> &&
    HasSupport<T>
```

9.33.2 Detailed Description

Concept for term types that are spatial terms. A spatial term is a term that has a functional space (HasFunctionalSpace) and a support (HasSupport).

9.34 serif::discretization::blocks::Term Concept Reference

Concept for term types in the discretization framework. To be a term a type must derive from term_base, must have a field_type member, and that field_type member must be a valid field.

#include <concepts.hpp>

9.34.1 Concept definition

```
template<typename T>
concept serif::discretization::blocks::Term =  std::derived_from<std::remove_cvref_t<T>, term_base> &&
    requires {typename std::remove_cvref_t<T>::field_type;} &&
    Field<typename std::remove_cvref_t<T>::field_type>
```

9.34.2 Detailed Description

Concept for term types in the discretization framework. To be a term a type must derive from term_base, must have a field_type member, and that field_type member must be a valid field.

9.35 serif::discretization::domain::BoundaryIsIdentifiedBy Concept Reference

#include <concepts.hpp>

9.35.1 Concept definition

```
template<typename BoundaryT, typename BoundaryIDT>
concept serif::discretization::domain::BoundaryIsIdentifiedBy =
    utils::misc::std_helper::SameType<BoundaryT, typename BoundaryIDT::boundary_type>
```

9.36 serif::discretization::domain::DomainIsIdentifiedBy Concept Reference

#include <concepts.hpp>

9.36.1 Concept definition

```
template<typename DomainT, typename DomainIDT>
concept serif::discretization::domain::DomainIsIdentifiedBy =
    utils::misc::std_helper::SameType<DomainT, typename DomainIDT::domain_type>
```

9.37 serif::discretization::domain::ids::IsBoundaryID Concept Reference

Concept for types that are boundary IDs in the discretization framework. To be a boundary ID a type must satisfy the is_boundary_id_v trait.

#include <concepts.hpp>

9.37.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::ids::IsBoundaryID = is_boundary_id_v<T>
```

9.37.2 Detailed Description

Concept for types that are boundary IDs in the discretization framework. To be a boundary ID a type must satisfy the is_boundary_id_v trait.

9.38 serif::discretization::domain::ids::IsDomainID Concept Reference

Concept for types that are domain IDs in the discretization framework. To be a domain ID a type must satisfy the is_domain_id_v trait.

#include <concepts.hpp>

9.38.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::ids::IsDomainID = is_domain_id_v<T>
```

9.38.2 Detailed Description

Concept for types that are domain IDs in the discretization framework. To be a domain ID a type must satisfy the is_domain_id_v trait.

9.39 serif::discretization::domain::ids::IsDomainOrBoundaryID Concept Reference

Concept for types that are either domain IDs or boundary IDs in the discretization framework.

#include <concepts.hpp>

9.39.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::ids::IsDomainOrBoundaryID = IsDomainID<T> || IsBoundaryID<T>
```

9.39.2 Detailed Description

Concept for types that are either domain IDs or boundary IDs in the discretization framework.

9.40 serif::discretization::domain::ids::lists::IsBoundaryIDList Concept Reference

Concept for types that are boundary ID lists in the discretization framework. To be a boundary ID list a type must satisfy the is_boundary_id_list_v trait.

#include <concepts.hpp>

9.40.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::ids::lists::IsBoundaryIDList = is_boundary_id_list_v<T>
```

9.40.2 Detailed Description

Concept for types that are boundary ID lists in the discretization framework. To be a boundary ID list a type must satisfy the is_boundary_id_list_v trait.

9.41 serif::discretization::domain::ids::lists::IsDomainIDList Concept Reference

Concept for types that are domain ID lists in the discretization framework. To be a domain ID list a type must satisfy the is_domain_id_list_v trait.

#include <concepts.hpp>

9.41.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::ids::lists::IsDomainIDList = is_domain_id_list_v<T>
```

9.41.2 Detailed Description

Concept for types that are domain ID lists in the discretization framework. To be a domain ID list a type must satisfy the is_domain_id_list_v trait.

9.42 serif::discretization::domain::ids::validation::AreDistinctBoundaryIDs Concept Reference

Concept to check if a pack of boundary IDs are distinct. This means that each ID is a valid boundary ID, no two IDs have the same integer value, and no two IDs refer to the same boundary type.

#include <concepts.hpp>

9.42.1 Concept definition

```
template<typename... BoundaryIDTs>
concept serif::discretization::domain::ids::validation::AreDistinctBoundaryIDs =
    (IsBoundaryID<BoundaryIDTs> && ...) &&
    HaveUniqueBoundaryIDs<BoundaryIDTs...> &&
    HaveUniqueBoundaries<BoundaryIDTs...>
```

9.42.2 Detailed Description

Concept to check if a pack of boundary IDs are distinct. This means that each ID is a valid boundary ID, no two IDs have the same integer value, and no two IDs refer to the same boundary type.

Template Parameters

BoundaryIDTs	The pack of boundary ID types to check.

9.43 serif::discretization::domain::ids::validation::AreDistinctDomainIDs Concept Reference

Concept to check if a pack of domain IDs are distinct. This means that each ID is a valid domain ID, no two IDs have the same integer value, and no two IDs refer to the same domain type.

#include <concepts.hpp>

9.43.1 Concept definition

```
template<typename... DomainIDTs>
concept serif::discretization::domain::ids::validation::AreDistinctDomainIDs =
        (IsDomainID<DomainIDTs> && ...) &&
        HaveUniqueDomainIDs<DomainIDTs...> &&
        HaveUniqueDomains<DomainIDTs...>
```

9.43.2 Detailed Description

Concept to check if a pack of domain IDs are distinct. This means that each ID is a valid domain ID, no two IDs have the same integer value, and no two IDs refer to the same domain type.

Template Parameters

DomainIDTs	The pack of domain ID types to check.

9.44 serif::discretization::domain::ids::validation::HaveUniqueBoundaries Concept Reference

Concept to check if a pack of boundary IDs have unique boundary types. This is used to ensure that no two boundary IDs in a list refer to the same boundary type.

#include <concepts.hpp>

9.44.1 Concept definition

```
template<typename... BoundaryIDTs>
concept serif::discretization::domain::ids::validation::HaveUniqueBoundaries =  AllTypesAreUnique<typename
        BoundaryIDTs::boundary_type...>
```

9.44.2 Detailed Description

Concept to check if a pack of boundary IDs have unique boundary types. This is used to ensure that no two boundary IDs in a list refer to the same boundary type.

Template Parameters

BoundaryIDTs	The pack of boundary ID types to check.

9.45 serif::discretization::domain::ids::validation::HaveUniqueBoundaryIDs Concept Reference

Concept to check if a pack of boundary IDs have unique ID values. This is used to ensure that no two boundary IDs in a list have the same ID.

#include <concepts.hpp>

9.45.1 Concept definition

```
template<typename... BoundaryIDTs>
concept serif::discretization::domain::ids::validation::HaveUniqueBoundaryIDs = all_unique_values(BoundaryIDTs::ID...)
```

9.45.2 Detailed Description

Concept to check if a pack of boundary IDs have unique ID values. This is used to ensure that no two boundary IDs in a list have the same ID.

Template Parameters

BoundaryIDTs	The pack of boundary ID types to check.

9.46 serif::discretization::domain::ids::validation::HaveUniqueDomainIDs Concept Reference

Concept to check if a pack of domain IDs have unique ID values. This is used to ensure that no two domain IDs in a list have the same ID.

#include <concepts.hpp>

9.46.1 Concept definition

```
template<typename... DomainIDTs>
concept serif::discretization::domain::ids::validation::HaveUniqueDomainIDs = all_unique_values(DomainIDTs::ID...)
```

9.46.2 Detailed Description

Concept to check if a pack of domain IDs have unique ID values. This is used to ensure that no two domain IDs in a list have the same ID.

Template Parameters

DomainIDTs	The pack of domain ID types to check.

9.47 serif::discretization::domain::ids::validation::HaveUniqueDomains Concept Reference

Concept to check if a pack of domain IDs have unique domain types. This is used to ensure that no two domain IDs in a list refer to the same domain type.

#include <concepts.hpp>

9.47.1 Concept definition

```
template<typename... DomainIDTs>
concept serif::discretization::domain::ids::validation::HaveUniqueDomains =  AllTypesAreUnique<typename
      DomainIDTs::domain_type...>
```

9.47.2 Detailed Description

Concept to check if a pack of domain IDs have unique domain types. This is used to ensure that no two domain IDs in a list refer to the same domain type.

Template Parameters

DomainIDTs	The pack of domain ID types to check.

9.48 serif::discretization::domain::IsBoundary Concept Reference

#include <concepts.hpp>

9.48.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::IsBoundary =  AIsBaseClassOfB<T, Boundary>
```

9.49 serif::discretization::domain::IsDomain Concept Reference

#include <concepts.hpp>

9.49.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::IsDomain = AIsBaseClassOfB<T, Domain>
```

9.50 serif::discretization::domain::IsDomainOrSet Concept Reference

#include <concepts.hpp>

9.50.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::IsDomainOrSet = IsDomain<T> || IsDomainSet<T>
```

9.51 serif::discretization::domain::IsDomainSet Concept Reference

#include <concepts.hpp>

9.51.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::IsDomainSet = is_domain_set_v<T>
```

9.52 serif::discretization::domain::relation::IsRelation Concept Reference

#include <concepts.hpp>

9.52.1 Concept definition

```
template<typename R>
concept serif::discretization::domain::relation::IsRelation = AIsBaseClassOfB<R, DomainRelation>
```

9.53 serif::discretization::domain::relation::lists::IsRelationList Concept Reference

#include <concepts.hpp>

9.53.1 Concept definition

```
template<typename RL>
concept serif::discretization::domain::relation::lists::IsRelationList = is_relation_list_v<RL>
```

9.54 serif::discretization::domain::relation::validation::HaveValidRelationEntities Concept Reference

#include <concepts.hpp>

9.54.1 Concept definition

```
template<typename DomainIDsT, typename BoundaryIDsT, typename RelationsT>
concept serif::discretization::domain::relation::validation::HaveValidRelationEntities =
        RelationsUseRegisteredEntities<DomainIDsT, BoundaryIDsT, RelationsT>::value
```

9.55 serif::discretization::domain::schema::IsSchema Concept Reference

#include <concepts.hpp>

9.55.1 Concept definition

```
template<typename T>
concept serif::discretization::domain::schema::IsSchema =  is_schema_v<T>
```

9.56 serif::discretization::forms::Form Concept Reference

Concept for types that are forms in the discretization framework. In order satisfy this concept a type must derive from the form base class and must define a member type called operand_types which is intended to be a type list of the trial and test terms for the form.

#include <concept.hpp>

9.56.1 Concept definition

```
template<typename FormCandidate>
concept serif::discretization::forms::Form =  std::derived_from<std::remove_cvref_t<FormCandidate>, form> &&
       requires {
          typename std::remove_cvref_t<FormCandidate>::operand_types;
       }
```

9.56.2 Detailed Description

Concept for types that are forms in the discretization framework. In order satisfy this concept a type must derive from the form base class and must define a member type called operand_types which is intended to be a type list of the trial and test terms for the form.

Todo Consider strengthening this concept to enforce that the operand_types type list contains only operand types and that there is an entry for TrialT and TestT. This would require additional compile-time checks to ensure that the types in the operand_types list are valid operand types and that they include the required trial and test terms.

9.57 serif::discretization::forms::IsOperand Concept Reference

Concept to validate a valid operand. An operand is valid if the following conditions are satisfied.

#include <operands.hpp>

9.57.1 Concept definition

```
template<typename OperandCandidate>
concept serif::discretization::forms::IsOperand =  requires {
          typename std::remove_cvref_t<OperandCandidate>::term_type;
          typename std::remove_cvref_t<OperandCandidate>::operation_type; } &&
       blocks::Term<typename std::remove_cvref_t<OperandCandidate>::term_type> &&
       IsOperation<typename std::remove_cvref_t<OperandCandidate>::operation_type>
```

9.57.2 Detailed Description

Concept to validate a valid operand. An operand is valid if the following conditions are satisfied.

- it has a member term_type
- it has a member operation_type
- term_type satisfies the Term concept (e.g. is a valid term)
- operation_type satisfies the IsOperation concept (e..g is a valid operation)

9.58 serif::discretization::forms::IsOperation Concept Reference

#include <operands.hpp>

9.58.1 Concept definition

```
template<typename OperationCandidate>
concept serif::discretization::forms::IsOperation =
       std::same_as<std::remove_cvref_t<OperationCandidate>, Operation::Value> ||
       std::same_as<std::remove_cvref_t<OperationCandidate>, Operation::Gradient> ||
       std::same_as<std::remove_cvref_t<OperationCandidate>, Operation::Divergence> ||
       std::same_as<std::remove_cvref_t<OperationCandidate>, Operation::Curl> ||
       std::same_as<std::remove_cvref_t<OperationCandidate>, Operation::NormalTrace>
```

9.58.2 Detailed Description

@breif Concept to validate that sone type is a valid operation, if you extend Operation you should extend this concept.

Todo Could C++26 reflection make keeping this and Operation parallel easier / more automatic since this code would live in the same translation unit?

9.59 serif::discretization::functional_space::IsSpaceDescriptor Concept Reference

Concept to validate a valid space descriptor. A space descriptor is valid if it is of type SpaceDescriptor. This is a very simple concept that just checks that the type is exactly SpaceDescriptor.

#include <space.hpp>

9.59.1 Concept definition

```
template<typename SpaceDescriptorCandidate>
concept serif::discretization::functional_space::IsSpaceDescriptor = std::is_same_v<SpaceDescriptor, SpaceDescriptorCandidate>
```

9.59.2 Detailed Description

Concept to validate a valid space descriptor. A space descriptor is valid if it is of type SpaceDescriptor. This is a very simple concept that just checks that the type is exactly SpaceDescriptor.

9.60 serif::discretization::quadrature::backend::mfem::Configurable↩ Integrator Concept Reference

#include <concepts.hpp>

9.60.1 Concept definition

```
template<typename IntegratorT>
concept serif::discretization::quadrature::backend::mfem::ConfigurableIntegrator = HasReferenceSetIntegrationRule<IntegratorT> ||
    HasPointerSetIntegrationRule<IntegratorT> ||
    HasSetIntRule<IntegratorT>
```

9.61 serif::discretization::quadrature::backend::mfem::HasPointerSet↩ IntegrationRule Concept Reference

#include <concepts.hpp>

9.61.1 Concept definition

```
template<typename IntegratorT>
concept serif::discretization::quadrature::backend::mfem::HasPointerSetIntegrationRule = requires (IntegratorT &integrator, const
     ::mfem::IntegrationRule &integration_rule) {
    integrator.SetIntRule(&integration_rule);
}
```

9.62 serif::discretization::quadrature::backend::mfem::HasReferenceSetIntegrationRule Concept Reference

#include <concepts.hpp>

9.62.1 Concept definition

```
template<typename IntegratorT>
concept serif::discretization::quadrature::backend::mfem::HasReferenceSetIntegrationRule =  requires (IntegratorT &integrator, const
        ::mfem::IntegrationRule &integration_rule) {
      integrator.SetIntegrationRule(integration_rule);
   }
```

9.63 serif::discretization::quadrature::backend::mfem::HasSetIntRule Concept Reference

#include <concepts.hpp>

9.63.1 Concept definition

```
template<typename IntegratorT>
concept serif::discretization::quadrature::backend::mfem::HasSetIntRule =  requires (IntegratorT &integrator, const
        ::mfem::IntegrationRule &integration_rule) {
      integrator.SetIntRule(integration_rule);
   }
```

9.64 serif::eos::models::BarotropicClosureEOS Concept Reference

#include <concepts.hpp>

9.64.1 Concept definition

```
template<typename EOSCandidate>
concept serif::eos::models::BarotropicClosureEOS =  EOSModel<EOSCandidate> &&
      EOSSupportingRelation<EOSCandidate, relations::DensityFromSpecificEnthalpy> &&
      EOSSupportingPartialDerivative<EOSCandidate, relations::DensityFromSpecificEnthalpy, dimensions::SpecificEnthalpy>
```

9.65 serif::eos::models::EOSModel Concept Reference

#include <concepts.hpp>

9.65.1 Concept definition

```
template<typename EOSCandidate>
concept serif::eos::models::EOSModel =  is_equation_of_state_model_v<EOSCandidate>
```

9.66 serif::eos::models::EOSSupportingPartialDerivative Concept Reference

#include <concepts.hpp>

9.66.1 Concept definition

```
template<typename EOS, typename RelationT, typename InputQuantity>
concept serif::eos::models::EOSSupportingPartialDerivative = supports_partial_derivative_v<EOS, RelationT, InputQuantity>
```

9.67 serif::eos::models::EOSSupportingRelation Concept Reference

#include <concepts.hpp>

9.67.1 Concept definition

```
template<typename EOS, typename RelationT>
concept serif::eos::models::EOSSupportingRelation = supports_relation_v<EOS, RelationT>
```

9.68 serif::eos::models::PressureForceEOS Concept Reference

#include <concepts.hpp>

9.68.1 Concept definition

```
template<typename EOSCandidate>
concept serif::eos::models::PressureForceEOS = EOSModel<EOSCandidate> &&
    EOSSupportingRelation<EOSCandidate, relations::PressureFromSpecificEnthalpy> &&
    EOSSupportingPartialDerivative<EOSCandidate, relations::PressureFromSpecificEnthalpy, dimensions::SpecificEnthalpy>
```

9.69 serif::eos::models::StructureSeedEOS Concept Reference

#include <concepts.hpp>

9.69.1 Concept definition

```
template<typename EOSCandidate>
concept serif::eos::models::StructureSeedEOS = EOSModel<EOSCandidate> &&
    EOSSupportingRelation<EOSCandidate, relations::SpecificEnthalpyFromDensity>
```

9.70 serif::eos::relations::CatalogContainingRelation Concept Reference

#include <concepts.hpp>

9.70.1 Concept definition

```
template<typename Catalog, typename RelationT>
concept serif::eos::relations::CatalogContainingRelation =  relation_catalog_contains_relation_v<Catalog, RelationT>
```

9.71 serif::eos::relations::RelationContainingInput Concept Reference

#include <concepts.hpp>

9.71.1 Concept definition

```
template<typename RelationT, typename Quantity>
concept serif::eos::relations::RelationContainingInput =  relation_contains_input_v<RelationT, Quantity>
```

9.72 serif::eos::relations::ValidRelationCatalog Concept Reference

#include <concepts.hpp>

9.72.1 Concept definition

```
template<typename CandidateCatalog>
concept serif::eos::relations::ValidRelationCatalog =  is_valid_relation_catalog_v<CandidateCatalog>
```

9.73 serif::eos::relations::ValidThermodynamicRelation Concept Reference

#include <concepts.hpp>

9.73.1 Concept definition

```
template<typename RelationCandidate>
concept serif::eos::relations::ValidThermodynamicRelation =  is_thermodynamic_relation_v<RelationCandidate>
```

9.74 serif::eos::runtime::RuntimeEOSModel Concept Reference

#include <concepts.hpp>

9.74.1 Concept definition

```
template<typename Candidate>
concept serif::eos::runtime::RuntimeEOSModel =  eos::models::EOSModel<Candidate> &&
      RuntimeCatalogIsSupported<typename std::remove_cvref_t<Candidate>::Relations>::value
```

9.75 serif::equations::Equation Concept Reference

Physics authoring contract; contributions share one residual row but may use different tests/supports.

#include <concepts.hpp>

9.75.1 Concept definition

```
template<typename T>
concept serif::equations::Equation =  requires {
        typename T::Test;
        { T::name } -> std::convertible_to<std::string_view>;
    } && ValidContributions<typename T::Test, ContributionsOf<T»::value
```

9.75.2 Detailed Description

Physics authoring contract; contributions share one residual row but may use different tests/supports.

9.76 serif::equations::PointEquation Concept Reference

#include <concepts.hpp>

9.76.1 Concept definition

```
template<typename T>
concept serif::equations::PointEquation =
        requires {
            typename T::Test;
            typename T::Operands;
        } && (discretization::blocks::SpatialTerm<typename T::Test> || discretization::blocks::GlobalScalarTerm<typename T::Test>)
         &&
        discretization::blocks::HasResidualBlock<typename T::Test> &&
        discretization::forms::IsOperand<TestOperandOf<T>> && std::same_as<typename TestOperandOf<T>::term_type,
         typename T::Test> &&
        requires(const T& equation, const EquationPoint<typename T::Operands>& point) {
            {
                equation.evaluate(point)
            } -> std::same_as<PointLinearization<typename T::Operands, discretization::forms::OperandValue<TestOperandOf<T»»;
        }
```

9.77 serif::geometry::deformation::DeformationMap Concept Reference

#include <actions.hpp>

9.77.1 Concept definition

```
template<typename Map>
concept serif::geometry::deformation::DeformationMap =  LinearDeformationMap<Map> || DifferentiableDeformationMap<Map>
```

9.78 serif::geometry::deformation::DifferentiableDeformationMap Concept Reference

Nonlinear maps explicitly provide every term needed by a Newton force pullback.

#include <actions.hpp>

9.78.1 Concept definition

```
template<typename Map>
concept serif::geometry::deformation::DifferentiableDeformationMap =  requires(const Map& map, std::span<const double> state,
       std::span<const double> direction, std::span<const double> dual, std::span<double> output, typename Map::Workspace&
       workspace) {
      typename Map::InputSpace;
      typename Map::OutputSpace;
      { map.input_size() } -> std::same_as<std::size_t>;
      { map.output_size() } -> std::same_as<std::size_t>;
      { map.make_workspace() } -> std::same_as<typename Map::Workspace>;
      { map.build_displacement(state, output, workspace) } -> std::same_as<DeformationStatus>;
      { map.apply_jacobian(state, direction, output, workspace) } -> std::same_as<DeformationStatus>;
      { map.apply_jacobian_transpose(state, dual, output, workspace) } -> std::same_as<DeformationStatus>;
      { map.apply_pullback_derivative(state, direction, dual, output, workspace) } -> std::same_as<DeformationStatus>;
   }
```

9.78.2 Detailed Description

Nonlinear maps explicitly provide every term needed by a Newton force pullback.

9.79 serif::geometry::deformation::LinearDeformationMap Concept Reference

Fixed maps supply their forward and work-conjugate transpose once.

#include <actions.hpp>

9.79.1 Concept definition

```
template<typename Map>
concept serif::geometry::deformation::LinearDeformationMap =  requires(const Map& map, std::span<const double> input,
       std::span<double> output) {
      typename Map::InputSpace;
      typename Map::OutputSpace;
      requires Map::is_linear;
      { map.input_size() } -> std::same_as<std::size_t>;
      { map.output_size() } -> std::same_as<std::size_t>;
      { map.apply(input, output) } -> std::same_as<DeformationStatus>;
      { map.apply_transpose(input, output) } -> std::same_as<DeformationStatus>;
   }
```

9.79.2 Detailed Description

Fixed maps supply their forward and work-conjugate transpose once.

9.80 serif::geometry::surface::DirectPressureSurfaceCompilable Concept Reference

A direct pressure residual requires an explicitly chosen pressure relation and an input quantity whose equation row it replaces.

#include <compilation.hpp>

9.80.1 Concept definition

```
template<typename EOS, typename Bindings, typename RowQuantity, typename Relation>
concept serif::geometry::surface::DirectPressureSurfaceCompilable =  eos::models::EOSModel<EOS> &&
     SurfaceStateBindings<Bindings> && eos::relations::ValidThermodynamicRelation<Relation> &&
    eos::relations::CatalogContainingRelation<typename std::remove_cvref_t<EOS>::Relations, Relation> &&
    std::same_as<typename Relation::OutputQuantity, dimensions::Pressure> &&
     eos::relations::RelationContainingInput<Relation, RowQuantity> &&
    SurfaceResidualRow<Bindings, RowQuantity> &&
    utils::dependencies::SatisfiesKeyList<Bindings, typename detail::RelationQuantities<Relation>::AllInputs> &&
    detail::supports_partials<EOS, Relation>(typename detail::RelationQuantities<Relation>::AllInputs{})
```

9.80.2 Detailed Description

A direct pressure residual requires an explicitly chosen pressure relation and an input quantity whose equation row it replaces.

9.81 serif::geometry::surface::IsobaricSurfaceType Concept Reference

#include <isobaric.hpp>

9.81.1 Concept definition

```
template<typename Candidate>
concept serif::geometry::surface::IsobaricSurfaceType =  std::same_as<std::remove_cvref_t<Candidate>, IsobaricSurface>
```

9.82 serif::geometry::surface::PressureSurfaceCompilable Concept Reference

Whether the available state selects exactly one pressure-to-carrier EOS relation.

#include <compilation.hpp>

9.82.1 Concept definition

```
template<typename EOS, typename Bindings, typename Carrier void>
concept serif::geometry::surface::PressureSurfaceCompilable =  eos::models::EOSModel<EOS> && SurfaceStateBindings<Bindings>
    &&
    (detail::MatchingPressureRelations<std::remove_cvref_t<EOS>, std::remove_cvref_t<Bindings>, Carrier, typename
     std::remove_cvref_t<EOS>::Relations>::count == 1)
```

9.82.2 Detailed Description

Whether the available state selects exactly one pressure-to-carrier EOS relation.

Template Parameters

Carrier	Optional physical quantity whose equation enforces the boundary condition.

Note

An ambiguous catalog never selects its first entry. Specifying a carrier narrows the same search.

9.83 serif::geometry::surface::PressureSurfaceFormulationType Concept Reference

#include <compilation.hpp>

9.83.1 Concept definition

```
template<typename Candidate>
concept serif::geometry::surface::PressureSurfaceFormulationType =  detail::is_pressure_surface_formulation<Candidate>
```

9.84 serif::geometry::surface::SurfaceResidualRow Concept Reference

The quantity enforcing the boundary must have an actual equation row, not just sampled values.

#include <relations.hpp>

9.84.1 Concept definition

```
template<typename Bindings, typename Quantity>
concept serif::geometry::surface::SurfaceResidualRow =  utils::dependencies::BindingsWithBinding<Bindings, Quantity> &&
      discretization::blocks::HasResidualBlock<utils::dependencies::bound_type_t<Bindings, Quantity>>
```

9.84.2 Detailed Description

The quantity enforcing the boundary must have an actual equation row, not just sampled values.

9.85 serif::geometry::surface::SurfaceScalarTerm Concept Reference

Scalar thermodynamic data may come from any declared scalar space; boundary enforcement is a separate backend capability.

#include <relations.hpp>

9.85.1 Concept definition

```
template<typename Term>
concept serif::geometry::surface::SurfaceScalarTerm =  discretization::blocks::Term<Term> && requires {
      typename discretization::forms::OperandValue<discretization::forms::Operand<Term>>;
      requires std::same_as<discretization::forms::OperandValue<discretization::forms::Operand<Term», double>;
   }
```

9.85.2 Detailed Description

Scalar thermodynamic data may come from any declared scalar space; boundary enforcement is a separate backend capability.

9.86 serif::geometry::surface::SurfaceStateBindings Concept Reference

Reuse dependency bindings, retaining distinct quantity roles even when they refer to one term.

#include <relations.hpp>

9.86.1 Concept definition

```
template<typename Bindings>
concept serif::geometry::surface::SurfaceStateBindings =  utils::dependencies::IsBindings<Bindings> &&
      detail::AreSurfaceBindings<std::remove_cvref_t<Bindings>>::value
```

9.86.2 Detailed Description

Reuse dependency bindings, retaining distinct quantity roles even when they refer to one term.

9.87 serif::operators::backend::mfem::AffinePointEquation Concept Reference

An affine scalar point condition supplies the two off-diagonal border blocks.

#include <nodal_data.hpp>

9.87.1 Concept definition

```
template<typename EquationT>
concept serif::operators::backend::mfem::AffinePointEquation =  NodalValueEquation<EquationT> &&
      requires {
         typename EquationT::Border;
         EquationT::reference_position;
      } &&
      discretization::blocks::SpatialTermTraits<typename EquationT::Test>::rank == 0 &&
      discretization::blocks::GlobalScalarTerm<typename EquationT::Border> && EquationT::affine
```

9.87.2 Detailed Description

An affine scalar point condition supplies the two off-diagonal border blocks.

9.88 serif::operators::backend::mfem::ClosureInputsAccept Concept Reference

9.88.1 Concept definition

```
template<typename DisplacementT>
concept serif::operators::backend::mfem::ClosureInputsAccept =  requires(
        const VolumeMesh& discretization,
        const VolumeCoefficients& coefficients,
        DisplacementT&& displacement
    ) {
        make_volume_inputs<Closure>(discretization, coefficients, std::forward<DisplacementT>(displacement));
    }
```

9.89 serif::operators::backend::mfem::ClosureResult Concept Reference

9.89.1 Concept definition

```
template<typename KeyT>
concept serif::operators::backend::mfem::ClosureResult =  requires(const VolumeCache& cache) {
        cache.template get<KeyT>();
    }
```

9.90 serif::operators::backend::mfem::EvaluableOperand Concept Reference

Check the complete interface used by preparation, Jacobian application, and its transpose.

#include <operand_evaluation.hpp>

9.90.1 Concept definition

```
template<typename OperandT>
concept serif::operators::backend::mfem::EvaluableOperand =  requires(
        const discretization::backend::mfem::DofMap& binding,
        const ::mfem::IntegrationRule& rule,
        ::mfem::ElementTransformation& transformation,
        const BasisEvaluation& reference_basis,
        BasisEvaluation& mapped_basis,
        TensorBasisCache& cache,
        BasisWorkspace& workspace,
        const ::mfem::Vector& coefficients,
        ::mfem::Vector& values,
        const typename OperandEvaluator<OperandT>::ReferenceValue& reference_value,
        const mapping::MappingPointContext& context,
        const mapping::MappingPointVariation& variation
    ) {
        typename OperandEvaluator<OperandT>::Value;
        typename OperandEvaluator<OperandT>::ReferenceValue;
        { OperandEvaluator<OperandT>::components } -> std::convertible_to<int>;
        { OperandEvaluator<OperandT>::reference_components } -> std::convertible_to<int>;
        { OperandEvaluator<OperandT>::geometry_dependent } -> std::convertible_to<bool>;
```

```
    { OperandEvaluator<OperandT>::sample_basis(binding, 0, rule, mapped_basis, cache, ContractionMode::automatic,
     transformation) } -> std::same_as<void>;
    { OperandEvaluator<OperandT>::evaluate(reference_basis, coefficients, values, workspace) } -> std::same_as<void>;
    { OperandEvaluator<OperandT>::integrate(reference_basis, coefficients, values, workspace) } -> std::same_as<void>;
    { OperandEvaluator<OperandT>::reference_value(coefficients, 0) } -> std::same_as<typename
     OperandEvaluator<OperandT>::ReferenceValue>;
    { OperandEvaluator<OperandT>::physical_value(reference_value, context) } -> std::same_as<typename
     OperandEvaluator<OperandT>::Value>;
    { OperandEvaluator<OperandT>::geometry_variation(reference_value, context, variation) } -> std::same_as<typename
     OperandEvaluator<OperandT>::Value>;
    { OperandEvaluator<OperandT>::map_basis_point(0, context, reference_basis, mapped_basis) } -> std::same_as<void>;
    { OperandEvaluator<OperandT>::map_basis_point(0, context, reference_basis, mapped_basis, &variation) } ->
     std::same_as<void>;
}
```

9.90.2 Detailed Description

Check the complete interface used by preparation, Jacobian application, and its transpose.

9.91 serif::operators::CanSubtract Concept Reference

9.91.1 Concept definition

```
template<typename LeftT, typename RightT>
concept serif::operators::CanSubtract =  requires { typename equations::EquationDifference<LeftT, RightT>; }
```

9.92 serif::operators::CanSum Concept Reference

9.92.1 Concept definition

```
template<typename LeftT, typename RightT>
concept serif::operators::CanSum =  requires { typename equations::EquationSum<LeftT, RightT>; }
```

9.93 serif::operators::NodalValueEquation Concept Reference

Nodal execution: an H1 value depending only on itself, with no explicit position dependence.

#include <concepts.hpp>

9.93.1 Concept definition

```
template<typename T>
concept serif::operators::NodalValueEquation =
    equations::Equation<T> &&
    equations::PointEquation<T> &&
    discretization::blocks::SpatialTerm<typename T::Test> &&
    discretization::blocks::SpatialTermTraits<typename T::Test>::rank <= 1 &&
    discretization::blocks::SpatialTermTraits<typename T::Test>::family == discretization::functional_space::Family::h1 &&
    std::same_as<equations::TestOperandOf<T>, discretization::forms::Operand<typename T::Test>> &&
    std::same_as<typename T::Operands, utils::types::TypeList<discretization::forms::Operand<typename T::Test>>> &&
    requires { requires T::position_independent; }
```

9.93.2 Detailed Description

Nodal execution: an H1 value depending only on itself, with no explicit position dependence.

9.94 serif::operators::ScalarValueOperand Concept Reference

#include <concepts.hpp>

9.94.1 Concept definition

```
template<typename OperandT>
concept serif::operators::ScalarValueOperand =
    discretization::forms::IsOperand<OperandT> &&
    ScalarValueTerm<typename OperandT::term_type> &&
    std::same_as<typename OperandT::operation_type, discretization::forms::Operation::Value>
```

9.95 serif::operators::ScalarValueTerm Concept Reference

#include <concepts.hpp>

9.95.1 Concept definition

```
template<typename TermT>
concept serif::operators::ScalarValueTerm =
    discretization::blocks::GlobalScalarTerm<TermT> ||
    (discretization::blocks::SpatialTerm<TermT> &&
    discretization::blocks::SpatialTermTraits<TermT>::rank == 0 &&
    (
        discretization::blocks::SpatialTermTraits<TermT>::family == discretization::functional_space::Family::h1 ||
        discretization::blocks::SpatialTermTraits<TermT>::family == discretization::functional_space::Family::l2
    ))
```

9.96 serif::operators::ScalarVolumeEquation Concept Reference

Equations executable by the current scalar-value volume operator.

#include <concepts.hpp>

9.96.1 Concept definition

```
template<typename EquationT>
concept serif::operators::ScalarVolumeEquation =
    equations::Equation<EquationT> &&
    ScalarValueTerm<typename EquationT::Test> &&
    HasScalarVolumeOperands<EquationT, typename EquationT::Operands>::value
```

9.96.2 Detailed Description

Equations executable by the current scalar-value volume operator.

9.97 serif::operators::VolumeEquation Concept Reference

One or more supported weak volume contributions assembled into a single residual row.

#include <concepts.hpp>

9.97.1 Concept definition

```
template<typename T>
concept serif::operators::VolumeEquation =  equations::Equation<T> &&
        VolumeContributions<equations::ContributionsOf<T>>::value
```

9.97.2 Detailed Description

One or more supported weak volume contributions assembled into a single residual row.

9.98 serif::operators::VolumeOperand Concept Reference

Mathematical operand combinations implemented by the current volume backend.

#include <concepts.hpp>

9.98.1 Concept definition

```
template<typename OperandT>
concept serif::operators::VolumeOperand =
       discretization::forms::IsOperand<OperandT> &&
       (
          ScalarValueOperand<OperandT> ||
          (
             discretization::blocks::SpatialTerm<typename OperandT::term_type> &&
             discretization::blocks::SpatialTermTraits<typename OperandT::term_type>::rank == 1 &&
             (
                discretization::blocks::SpatialTermTraits<typename OperandT::term_type>::family ==
       discretization::functional_space::Family::h1 ||
                discretization::blocks::SpatialTermTraits<typename OperandT::term_type>::family ==
       discretization::functional_space::Family::hdiv
             ) &&
             (
                std::same_as<typename OperandT::operation_type, discretization::forms::Operation::Value> ||
                std::same_as<typename OperandT::operation_type, discretization::forms::Operation::Divergence>
             )
          )
       )
```

9.98.2 Detailed Description

Mathematical operand combinations implemented by the current volume backend.

9.99 serif::utils::dependencies::BindingsWithBinding Concept Reference

#include <binding.hpp>

9.99.1 Concept definition

```
template<typename BindingsT, typename KeyT>
concept serif::utils::dependencies::BindingsWithBinding = IsBindings<BindingsT> && has_binding_v<BindingsT, KeyT>
```

9.100 serif::utils::dependencies::CanPrepare Concept Reference

#include <cache_traits.hpp>

9.100.1 Concept definition

```
template<typename GraphT, typename ContextT, typename ErrorT>
concept serif::utils::dependencies::CanPrepare = detail::ComputationCacheTraits<std::remove_cvref_t<GraphT>,
    std::remove_cvref_t<ContextT>, ErrorT>::valid
```

9.101 serif::utils::dependencies::CompatibleComputations Concept Reference

#include <computation_graph.hpp>

9.101.1 Concept definition

```
template<typename... Computations>
concept serif::utils::dependencies::CompatibleComputations =
    (IsComputation<Computations> && ...) &&
    misc::traits::all_unique_types_v<typename std::remove_cvref_t<Computations>::Key...> &&
    detail::Graph<std::remove_cvref_t<Computations>...>::acyclic()
```

9.102 serif::utils::dependencies::ComputationGraphFor Concept Reference

#include <schema.hpp>

9.102.1 Concept definition

```
template<typename GraphT, typename ContextT>
concept serif::utils::dependencies::ComputationGraphFor = IsComputationGraph<GraphT> && IsDependencyContext<ContextT>
    &&
    detail::ComputationGraphSchema<std::remove_cvref_t<GraphT>, std::remove_cvref_t<ContextT>>::valid
```

9.103 serif::utils::dependencies::IsBinding Concept Reference

#include <binding.hpp>

9.103.1 Concept definition

```
template<typename T>
concept serif::utils::dependencies::IsBinding =  requires {
        typename T::Key;
        typename T::Target;
    } && std::is_same_v<T, Binding<typename T::Key, typename T::Target»
```

9.104 serif::utils::dependencies::IsBindings Concept Reference

#include <binding.hpp>

9.104.1 Concept definition

```
template<typename T>
concept serif::utils::dependencies::IsBindings =  is_bindings_v<std::remove_cvref_t<T>>
```

9.105 serif::utils::dependencies::IsComputation Concept Reference

#include <computation.hpp>

9.105.1 Concept definition

```
template<typename T>
concept serif::utils::dependencies::IsComputation =  is_computation_v<std::remove_cvref_t<T>>
```

9.106 serif::utils::dependencies::IsComputationGraph Concept Reference

#include <computation_graph.hpp>

9.106.1 Concept definition

```
template<typename T>
concept serif::utils::dependencies::IsComputationGraph =  is_computation_graph_v<std::remove_cvref_t<T>>
```

9.107 serif::utils::dependencies::IsDependencyContext Concept Reference

#include <dependency_context.hpp>

9.107.1 Concept definition

```
template<typename T>
concept serif::utils::dependencies::IsDependencyContext =  is_dependency_context_v<std::remove_cvref_t<T>>
```

9.108 serif::utils::dependencies::IsKeyList Concept Reference

#include <key_list.hpp>

9.108.1 Concept definition

```
template<typename T>
concept serif::utils::dependencies::IsKeyList = is_key_list_v<std::remove_cvref_t<T>>
```

9.109 serif::utils::dependencies::IsOwnedComputation Concept Reference

#include <computation_store.hpp>

9.109.1 Concept definition

```
template<typename T>
concept serif::utils::dependencies::IsOwnedComputation = is_owned_computation_v<std::remove_cvref_t<T>>
```

9.110 serif::utils::dependencies::IsResource Concept Reference

#include <resource.hpp>

9.110.1 Concept definition

```
template<typename T>
concept serif::utils::dependencies::IsResource = is_resource_v<std::remove_cvref_t<T>>
```

9.111 serif::utils::dependencies::SatisfiesKeyList Concept Reference

#include <key_list.hpp>

9.111.1 Concept definition

```
template<typename BindingsT, typename KeysT>
concept serif::utils::dependencies::SatisfiesKeyList = IsBindings<BindingsT> && IsKeyList<KeysT> &&
    KeyListBindings<BindingsT, std::remove_cvref_t<KeysT>>::value
```

9.112 serif::utils::dependencies::SnapshotPolicyFor Concept Reference

#include <snapshot.hpp>

9.112.1 Concept definition

```
template<typename PolicyT, typename ValueT>
concept serif::utils::dependencies::SnapshotPolicyFor =  requires(const PolicyT& policy, const ValueT& value) {
        policy.capture(value);
        requires std::is_object_v<decltype(policy.capture(value))>;
        requires std::move_constructible<decltype(policy.capture(value))>;
        { policy.matches(std::declval<const decltype(policy.capture(value))&>(), value) } -> std::convertible_to<bool>;
    } && (!requires(const PolicyT& policy, const ValueT& value) { policy.identity(value); } ||
        requires(const PolicyT& policy, const ValueT& value) { { policy.identity(value) } -> std::convertible_to<const void*>; })
```

9.113 serif::utils::misc::concepts::AllTypesAreUnique Concept Reference

#include <type_uniqueness.hpp>

9.113.1 Concept definition

```
template<typename... Ts>
concept serif::utils::misc::concepts::AllTypesAreUnique =  traits::all_unique_types_v<Ts...>
```

9.114 serif::utils::misc::concepts::IsComparableToNumeric Concept Reference

#include <numeric.hpp>

9.114.1 Concept definition

```
template<typename T>
concept serif::utils::misc::concepts::IsComparableToNumeric =  std::totally_ordered_with<T, float> ||
        std::totally_ordered_with<T, double> ||
        std::totally_ordered_with<T, long double> ||
        std::totally_ordered_with<T, int> ||
        std::totally_ordered_with<T, long> ||
        std::totally_ordered_with<T, long long>
```

9.115 serif::utils::misc::concepts::IsEnum Concept Reference

#include <numeric.hpp>

9.115.1 Concept definition

```
template<typename EnumCandidate>
concept serif::utils::misc::concepts::IsEnum =  std::is_enum_v<EnumCandidate>
```

9.116 serif::utils::misc::concepts::IsException Concept Reference

#include <exceptions.hpp>

9.116.1 Concept definition

```
template<typename ExceptionCandidate>
concept serif::utils::misc::concepts::IsException =  std::is_base_of_v<std::exception, ExceptionCandidate>
```

9.117 serif::utils::misc::concepts::IsExceptionWithErrorCode Concept Reference

#include <exceptions.hpp>

9.117.1 Concept definition

```
template<typename ExceptionCandidate, typename ErrorCode>
concept serif::utils::misc::concepts::IsExceptionWithErrorCode =  IsException<ExceptionCandidate> &&
                                  std::constructible_from<ExceptionCandidate, ErrorCode, std::string>
```

9.118 serif::utils::misc::concepts::IsNumeric Concept Reference

#include <numeric.hpp>

9.118.1 Concept definition

```
template<typename T>
concept serif::utils::misc::concepts::IsNumeric =  std::integral<T> || std::floating_point<T>
```

9.119 serif::utils::misc::concepts::IsOrderable Concept Reference

#include <numeric.hpp>

9.119.1 Concept definition

```
template<typename T>
concept serif::utils::misc::concepts::IsOrderable =  std::totally_ordered<T>
```

9.120 serif::utils::misc::concepts::SetOfOneOrTwo Concept Reference

#include <numeric.hpp>

9.120.1 Concept definition

```
template<typename... Ts>
concept serif::utils::misc::concepts::SetOfOneOrTwo =  (sizeof...(Ts) == 1) || (sizeof...(Ts) == 2)
```

Chapter 10

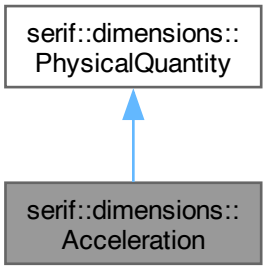
Class Documentation

10.1 serif::dimensions::Acceleration Struct Reference

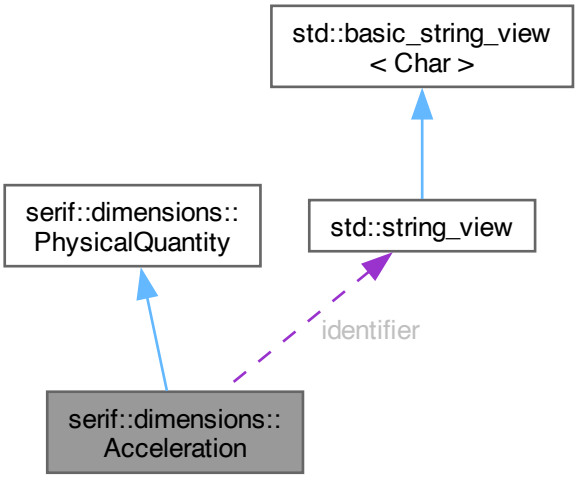
Dimension of Acceleration.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Acceleration:



Collaboration diagram for serif::dimensions::Acceleration:



Static Public Attributes

- static constexpr std::string_view identifier = "acceleration"

10.1.1 Detailed Description

Dimension of Acceleration.

10.1.2 Member Data Documentation

10.1.2.1 identifier

std::string_view serif::dimensions::Acceleration::identifier = "acceleration" [static], [constexpr]

The documentation for this struct was generated from the following file:

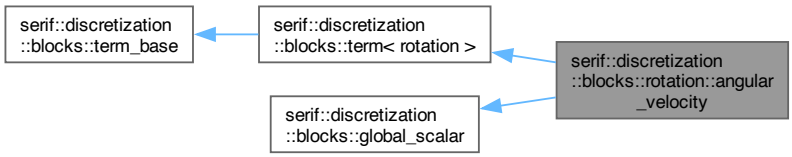
- src/include/serif/dimensions/dimensions.hpp

10.2 serif::discretization::blocks::rotation::angular_velocity Struct Reference

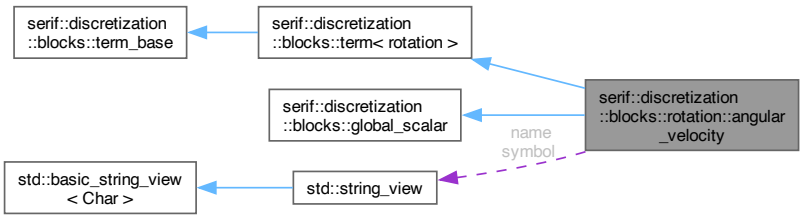
The angular_velocity term represents the angular velocity of the system. It is a global scalar that is associated with the rotation of the system. This is used to enforce the rotation condition in the weak form when we are in the rigid rotation case. Note that downstream the way that this currently works is to calculate the expected angular velocity given the current mass, geometry, density, and angular momentum and then prescribe, given those, what the angular velocity should be. Therefore, this does not enforce that the angular velocity must be constant over the entire solver run, but rather that the angular velocity is consistent with the current state of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::rotation::angular_velocity:



Collaboration diagram for serif::discretization::blocks::rotation::angular_velocity:



Public Types

- using Support = domain::StellarDomains

  The support of the angular velocity term is the stellar domains since we only want to enforce the rotation condition in the stellar domains.
- using Quantity = dimensions::AngularVelocity

  As we are in the rigid rotation case the angular velocity term is associated with the angular velocity. This will not generalize to the differential rotation case.

Public Types inherited from serif::discretization::blocks::term< rotation >

- using field_type

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<angular_velocity>()
  Compile time reflected name of the term.
- static constexpr std::string_view symbol = "Ω"
  Compile time symbol used for the term.

10.2.1 Detailed Description

The angular_velocity term represents the angular velocity of the system. It is a global scalar that is associated with the rotation of the system. This is used to enforce the rotation condition in the weak form when we are in the rigid rotation case. Note that downstream the way that this currently works is to calculate the expected angular velocity given the current mass, geometry, density, and angular momentum and then prescribe, given those, what the angular velocity should be. Therefore, this does not enforce that the angular velocity must be constant over the entire solver run, but rather that the angular velocity is consistent with the current state of the system.

10.2.2 Member Typedef Documentation

10.2.2.1 Quantity

using serif::discretization::blocks::rotation::angular_velocity::Quantity = dimensions::AngularVelocity

As we are in the rigid rotation case the angular velocity term is associated with the angular velocity. This will not generalize to the differential rotation case.

10.2.2.2 Support

using serif::discretization::blocks::rotation::angular_velocity::Support = domain::StellarDomains

The support of the angular velocity term is the stellar domains since we only want to enforce the rotation condition in the stellar domains.

10.2.3 Member Data Documentation

10.2.3.1 name

std::string_view serif::discretization::blocks::rotation::angular_velocity::name = meta::type_name<angular_velocity>() [static], [constexpr]

Compile time reflected name of the term.

10.2.3.2 symbol

std::string_view serif::discretization::blocks::rotation::angular_velocity::symbol = "Ω" [static], [constexpr]

Compile time symbol used for the term.

The documentation for this struct was generated from the following file:

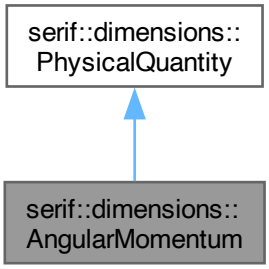
- src/include/serif/discretization/blocks/fields.hpp

10.3 serif::dimensions::AngularMomentum Struct Reference

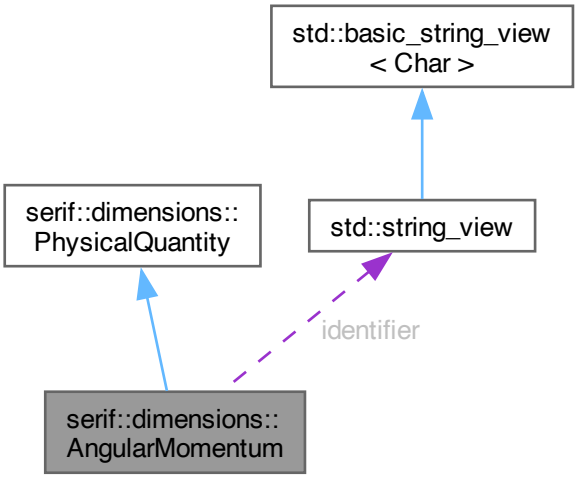
Dimension of Angular Momentum.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::AngularMomentum:



Collaboration diagram for serif::dimensions::AngularMomentum:



Static Public Attributes

- static constexpr std::string_view identifier = "angular_momentum"

10.3.1 Detailed Description

Dimension of Angular Momentum.

10.3.2 Member Data Documentation

10.3.2.1 identifier

std::string_view serif::dimensions::AngularMomentum::identifier = "angular_momentum" [static], [constexpr]

The documentation for this struct was generated from the following file:

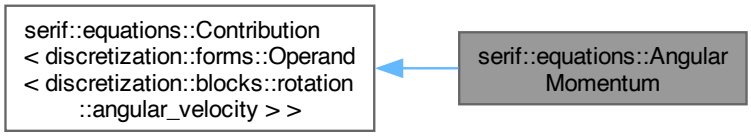
- src/include/serif/dimensions/dimensions.hpp

10.4 serif::equations::AngularMomentum Class Reference

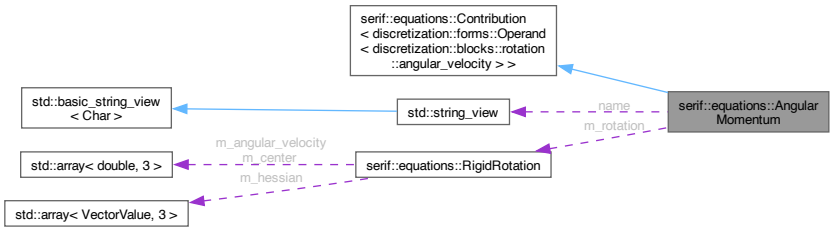
Close the unknown angular speed with integral_star rho Omega s^2 dV - target_momentum = 0.

#include <angular_momentum.hpp>

Inheritance diagram for serif::equations::AngularMomentum:



Collaboration diagram for serif::equations::AngularMomentum:



Public Types

- using Density = discretization::blocks::density::mass
- using AngularVelocity = discretization::blocks::rotation::angular_velocity
- using DensityValue = discretization::forms::Operand<Density>
- using AngularVelocityValue = discretization::forms::Operand<AngularVelocity>
- using Operands = utils::types::TypeList<DensityValue, AngularVelocityValue>

Public Types inherited from serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::rotation::angular_v

- using Test
- using TestOperand
- using Support

Public Member Functions

- AngularMomentum (double target_momentum, RigidRotation::Vector axis={0.0, 0.0, 1.0}, RigidRotation::Vector center={})

  The target is nonnegative; signed angular speeds remain valid trial states.
- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const
- int additional_order (DensityValue, int position_order) const
- int additional_order (AngularVelocityValue, int position_order) const
- double residual_offset () const
- double validation_integrand (const EquationPoint< Operands > &, const PointLinearization< Operands > &physics) const

  dF/dOmega is rho s^2, including at zero angular speed.
- void validate_integral (double moment_of_inertia) const

  Validate the total moment, allowing signed trial densities with nonnegative inertia.

Static Public Attributes

- static constexpr std::string_view name = "angular momentum"

Private Attributes

- double m_target_momentum
- RigidRotation m_rotation

10.4.1 Detailed Description

Close the unknown angular speed with integral_star rho Omega s^2 dV - target_momentum = 0.

10.4.2 Member Typedef Documentation

10.4.2.1 AngularVelocity

using serif::equations::AngularMomentum::AngularVelocity = discretization::blocks::rotation::angular_velocity

10.4.2.2 AngularVelocityValue

using serif::equations::AngularMomentum::AngularVelocityValue = discretization::forms::Operand<AngularVelocity>

10.4.2.3 Density

using serif::equations::AngularMomentum::Density = discretization::blocks::density::mass

10.4.2.4 DensityValue

using serif::equations::AngularMomentum::DensityValue = discretization::forms::Operand<Density>

10.4.2.5 Operands

using serif::equations::AngularMomentum::Operands = utils::types::TypeList<DensityValue, AngularVelocityValue>

10.4.3 Constructor & Destructor Documentation

10.4.3.1 AngularMomentum()

serif::equations::AngularMomentum::AngularMomentum (
double target_momentum,
RigidRotation::Vector axis = {0.0, 0.0, 1.0},
RigidRotation::Vector center = {}) [inline], [explicit]

The target is nonnegative; signed angular speeds remain valid trial states.

10.4.4 Member Function Documentation

10.4.4.1 additional_order() [1/2]

int serif::equations::AngularMomentum::additional_order (
AngularVelocityValue ,
int position_order) const [inline], [nodiscard]

10.4.4.2 additional_order() [2/2]

int serif::equations::AngularMomentum::additional_order (
DensityValue ,
int position_order) const [inline], [nodiscard]

10.4.4.3 evaluate()

PointLinearization< Operands > serif::equations::AngularMomentum::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

10.4.4.4 residual_offset()

double serif::equations::AngularMomentum::residual_offset () const [inline], [nodiscard]

10.4.4.5 validate_integral()

void serif::equations::AngularMomentum::validate_integral (
double moment_of_inertia) const [inline]

Validate the total moment, allowing signed trial densities with nonnegative inertia.

10.4.4.6 validation_integrand()

double serif::equations::AngularMomentum::validation_integrand (
const EquationPoint< Operands > & ,
const PointLinearization< Operands > & physics) const [inline], [nodiscard]

dF/dOmega is rho s^2, including at zero angular speed.

10.4.5 Member Data Documentation

10.4.5.1 m_rotation

RigidRotation serif::equations::AngularMomentum::m_rotation [private]

10.4.5.2 m_target_momentum

double serif::equations::AngularMomentum::m_target_momentum [private]

10.4.5.3 name

std::string_view serif::equations::AngularMomentum::name = "angular momentum" [static], [constexpr]

The documentation for this class was generated from the following file:

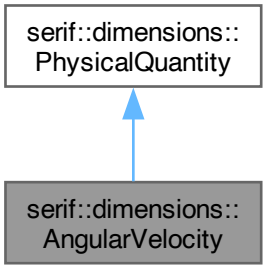
- src/include/serif/equations/angular_momentum.hpp

10.5 serif::dimensions::AngularVelocity Struct Reference

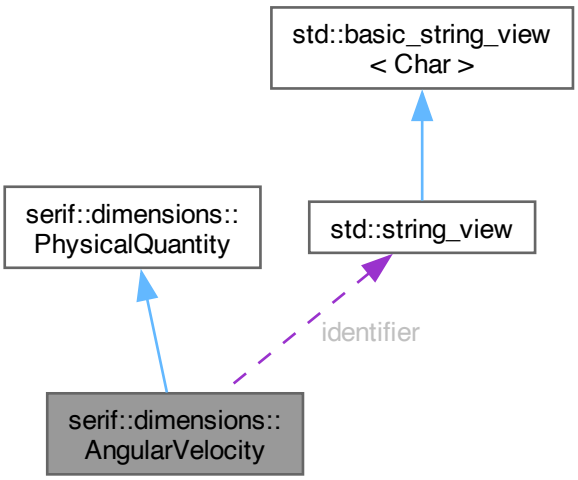
Dimension of Angular Velocity.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::AngularVelocity:



Collaboration diagram for serif::dimensions::AngularVelocity:



Static Public Attributes

- static constexpr std::string_view identifier = "angular_velocity"

10.5.1 Detailed Description

Dimension of Angular Velocity.

10.5.2 Member Data Documentation

10.5.2.1 identifier

std::string_view serif::dimensions::AngularVelocity::identifier = "angular_velocity" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.6 serif::utils::dependencies::detail::AppendBinding< BindingsT, BindingT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

10.7 serif::utils::dependencies::detail::AppendBinding< Bindings< BindingsTs... >, BindingT > Struct Template Reference

#include <schema.hpp>

Public Types

- using type = Bindings<BindingsTs..., BindingT>

10.7.1 Member Typedef Documentation

10.7.1.1 type

template<typename... BindingsTs, typename BindingT>
using serif::utils::dependencies::detail::AppendBinding< Bindings< BindingsTs... >, BindingT >::type = Bindings<Bindings↩
Ts..., BindingT>

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

10.8 serif::utils::dependencies::detail::AppendKeys< ResultT, Keys > Struct Template Reference

#include <type_lists.hpp>

Public Types

- using type = ResultT

10.8.1 Member Typedef Documentation

10.8.1.1 type

template<typename ResultT, typename... Keys>
using serif::utils::dependencies::detail::AppendKeys< ResultT, Keys >::type = ResultT

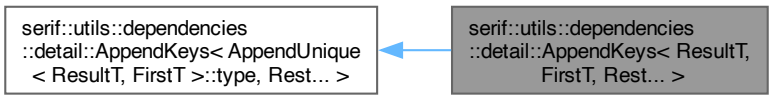
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/type_lists.hpp

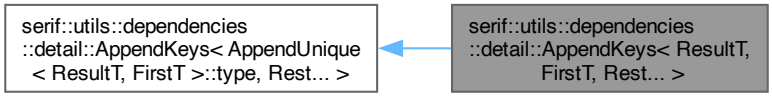
10.9 serif::utils::dependencies::detail::AppendKeys< ResultT, FirstT, Rest... > Struct Template Reference

#include <type_lists.hpp>

Inheritance diagram for serif::utils::dependencies::detail::AppendKeys< ResultT, FirstT, Rest... >:



Collaboration diagram for serif::utils::dependencies::detail::AppendKeys< ResultT, FirstT, Rest... >:



Public Types

- using type

Public Types inherited from
serif::utils::dependencies::detail::AppendKeys< AppendUnique< ResultT, FirstT >::type, Rest... >

- using type

10.9.1 Member Typedef Documentation

10.9.1.1 type

using serif::utils::dependencies::detail::AppendKeys< ResultT, Keys >::type

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/type_lists.hpp

10.10 serif::utils::dependencies::detail::AppendUnique< ListT, T > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/type_lists.hpp

10.11 serif::utils::dependencies::detail::AppendUnique< KeyList< Keys... >, T > Struct Template Reference

#include <type_lists.hpp>

Public Types

- using type = std::conditional_t<(std::is_same_v<T, Keys> || ...), KeyList<Keys...>, KeyList<Keys..., T>>

10.11.1 Member Typedef Documentation

10.11.1.1 type

template<typename... Keys, typename T>
using serif::utils::dependencies::detail::AppendUnique< KeyList< Keys... >, T >::type = std::conditional_t<(std::is_same_v<T, Keys> || ...), KeyList<Keys...>, KeyList<Keys..., T>>

The documentation for this struct was generated from the following file:

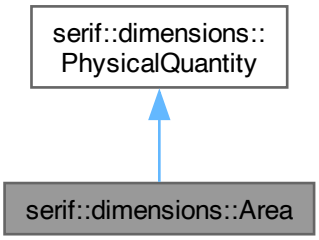
- src/include/serif/utils/dependencies/detail/type_lists.hpp

10.12 serif::dimensions::Area Struct Reference

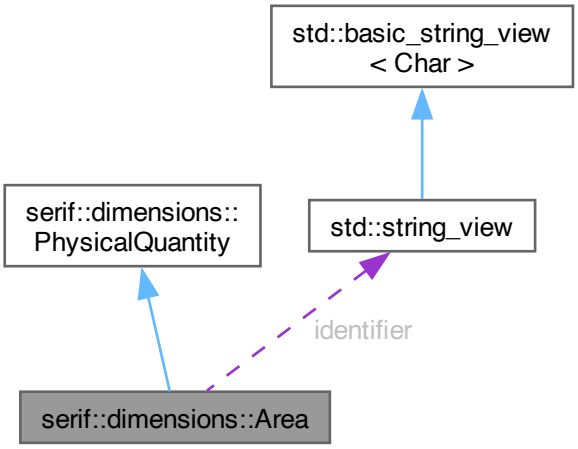
Dimension of Area.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Area:



Collaboration diagram for serif::dimensions::Area:



Static Public Attributes

- static constexpr std::string_view identifier = "area"

10.12.1 Detailed Description

Dimension of Area.

10.12.2 Member Data Documentation

10.12.2.1 identifier

std::string_view serif::dimensions::Area::identifier = "area" [static], [constexpr]

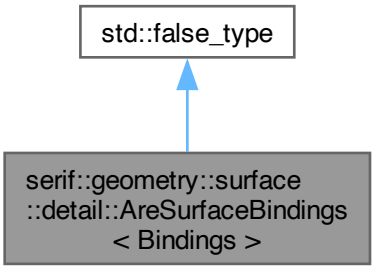
The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

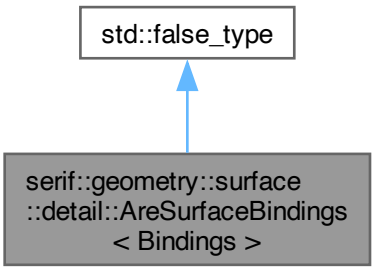
10.13 serif::geometry::surface::detail::AreSurfaceBindings< Bindings > Struct Template Reference

#include <relations.hpp>

Inheritance diagram for serif::geometry::surface::detail::AreSurfaceBindings< Bindings >:



Collaboration diagram for serif::geometry::surface::detail::AreSurfaceBindings< Bindings >:



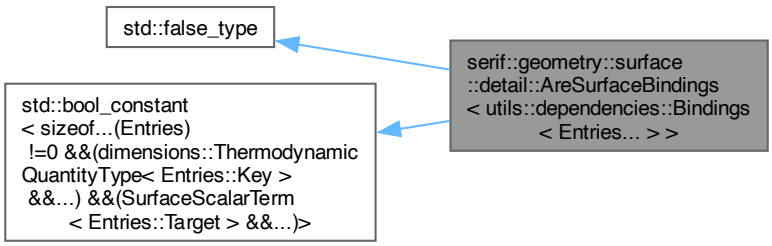
The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/detail/relations.hpp

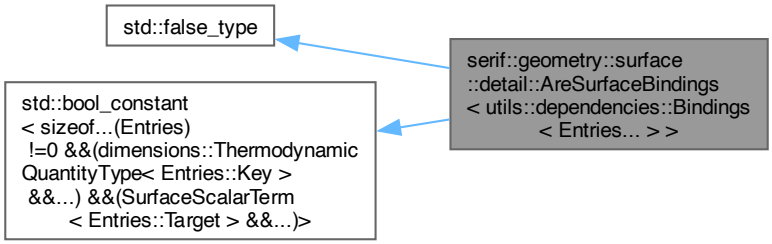
10.14 serif::geometry::surface::detail::AreSurfaceBindings< utils::dependencies::Bindings< Entries... > > Struct Template Reference

#include <relations.hpp>

Inheritance diagram for serif::geometry::surface::detail::AreSurfaceBindings< utils::dependencies::Bindings< Entries... > >:



Collaboration diagram for serif::geometry::surface::detail::AreSurfaceBindings< utils::dependencies::Bindings< Entries... > >:



The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/detail/relations.hpp

10.15 serif::operators::detail::AsKeyList< ListT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

10.16 serif::operators::detail::AsKeyList< utils::types::TypeList< T... > > Struct Template Reference

#include <plan.hpp>

Public Types

- using type = utils::dependencies::CombinedKeyList<utils::dependencies::KeyList<T>...>

10.16.1 Member Typedef Documentation

10.16.1.1 type

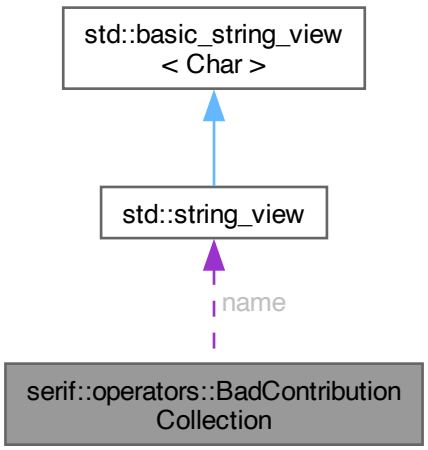
template<typename... T>
using serif::operators::detail::AsKeyList< utils::types::TypeList< T... > >::type = utils::dependencies::CombinedKeyList<utils::dependencies::KeyL

The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

10.17 serif::operators::BadContributionCollection Struct Reference

Collaboration diagram for serif::operators::BadContributionCollection:



Public Types

- using Test = Displacement

Public Member Functions

- auto contributions () const

Static Public Attributes

- static constexpr std::string_view name = "non-tuple contributions"

10.17.1 Member Typedef Documentation

10.17.1.1 Test

using serif::operators::BadContributionCollection::Test = Displacement

10.17.2 Member Function Documentation

10.17.2.1 contributions()

auto serif::operators::BadContributionCollection::contributions () const [inline]

10.17.3 Member Data Documentation

10.17.3.1 name

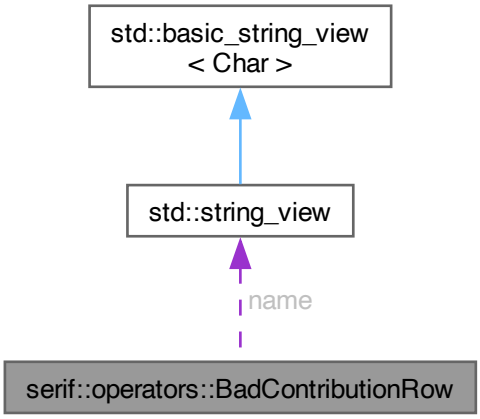
std::string_view serif::operators::BadContributionCollection::name = "non-tuple contributions" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/compile_time_checks/serif/operators/plan.cpp

10.18 serif::operators::BadContributionRow Struct Reference

Collaboration diagram for serif::operators::BadContributionRow:



Public Types

- using Test = Density

Public Member Functions

- auto contributions () const

  Bad user descriptions should fail the concept rather than instantiate invalid aliases.

Static Public Attributes

- static constexpr std::string_view name = "malformed contribution"

10.18.1 Member Typedef Documentation

10.18.1.1 Test

using serif::operators::BadContributionRow::Test = Density

10.18.2 Member Function Documentation

10.18.2.1 contributions()

auto serif::operators::BadContributionRow::contributions () const [inline]

Bad user descriptions should fail the concept rather than instantiate invalid aliases.

10.18.3 Member Data Documentation

10.18.3.1 name

std::string_view serif::operators::BadContributionRow::name = "malformed contribution" [static], [constexpr]

The documentation for this struct was generated from the following file:

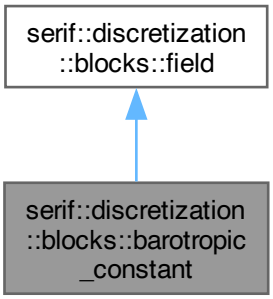
- src/compile_time_checks/serif/operators/plan.cpp

10.19 serif::discretization::blocks::barotropic_constant Struct Reference

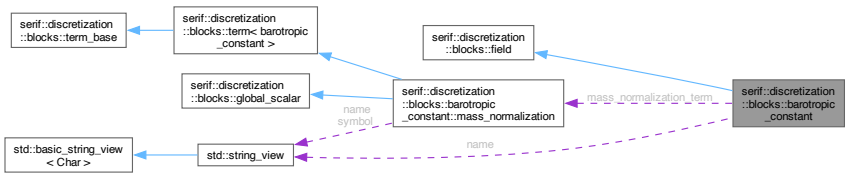
The barotropic_constant field is a field that represents the barotropic constant of the system. It is a global scalar that is associated with the mass normalization term. This is used to enforce the mass normalization condition in the weak form when we are in the barotropic case. This field will not generalize as useful to the non-barotropic case. Note that we use compile time reflection to get the name of the field and the terms.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::barotropic_constant:



Collaboration diagram for serif::discretization::blocks::barotropic_constant:



Classes

- struct mass_normalization

  the mass_normalization term represents the mass normalization condition of the system. That is that it enforces that the total mass of the system is equal to the target mass in a weak sense. This is both a term and a global scalar.

Public Types

- using Terms = utils::types::TypeList<mass_normalization>

  Type list of terms for compile time reflection and iteration over the terms of the field.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<barotropic_constant>()

  Compile time reflected name of the field.
- static constexpr mass_normalization mass_normalization_term {}

  Static constexpr instance of the mass_normalization term. This is used to access the term in the discretization.

10.19.1 Detailed Description

The barotropic_constant field is a field that represents the barotropic constant of the system. It is a global scalar that is associated with the mass normalization term. This is used to enforce the mass normalization condition in the weak form when we are in the barotropic case. This field will not generalize as useful to the non-barotropic case. Note that we use compile time reflection to get the name of the field and the terms.

10.19.2 Member Typedef Documentation

10.19.2.1 Terms

using serif::discretization::blocks::barotropic_constant::Terms = utils::types::TypeList<mass_normalization>

Type list of terms for compile time reflection and iteration over the terms of the field.

10.19.3 Member Data Documentation

10.19.3.1 mass_normalization_term

mass_normalization serif::discretization::blocks::barotropic_constant::mass_normalization_term {} [static], [constexpr]

Static constexpr instance of the mass_normalization term. This is used to access the term in the discretization.

10.19.3.2 name

std::string_view serif::discretization::blocks::barotropic_constant::name = meta::type_name<barotropic_constant>() [static], [constexpr]

Compile time reflected name of the field.

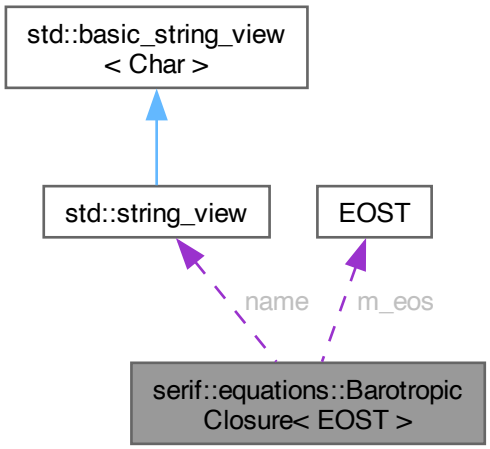
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

10.20 serif::equations::BarotropicClosure< EOST > Class Template Reference

#include <barotropic_closure.hpp>

Collaboration diagram for serif::equations::BarotropicClosure< EOST >:



Public Types

- using Test = discretization::blocks::density::mass
- using Enthalpy = discretization::blocks::enthalpy::specific
- using Operands = utils::types::TypeList<discretization::forms::Operand<Test>, discretization::forms::Operand<Entha

Public Member Functions

- BarotropicClosure (EOST equation_of_state, const double enthalpy_power)
- BarotropicClosure (EOST equation_of_state)
- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const
- template<typename OperandT>
  int additional_order (OperandT, int) const

Static Public Attributes

- static constexpr std::string_view name = "barotropic closure"

Private Attributes

- EOST m_eos
- double m_enthalpy_power

10.20.1 Member Typedef Documentation

10.20.1.1 Enthalpy

template<eos::models::BarotropicClosureEOS EOST>
using serif::equations::BarotropicClosure< EOST >::Enthalpy = discretization::blocks::enthalpy::specific

10.20.1.2 Operands

template<eos::models::BarotropicClosureEOS EOST>
using serif::equations::BarotropicClosure< EOST >::Operands = utils::types::TypeList<discretization::forms::Operand<Test>, discretization::forms::Operand<Enthalpy>>

10.20.1.3 Test

template<eos::models::BarotropicClosureEOS EOST>
using serif::equations::BarotropicClosure< EOST >::Test = discretization::blocks::density::mass

10.20.2 Constructor & Destructor Documentation

10.20.2.1 BarotropicClosure() [1/2]

template<eos::models::BarotropicClosureEOS EOST>
serif::equations::BarotropicClosure< EOST >::BarotropicClosure (
EOST equation_of_state,
const double enthalpy_power) [inline], [explicit]

10.20.2.2 BarotropicClosure() [2/2]

template<eos::models::BarotropicClosureEOS EOST>
serif::equations::BarotropicClosure< EOST >::BarotropicClosure (
EOST equation_of_state) [inline], [explicit]

10.20.3 Member Function Documentation

10.20.3.1 additional_order()

template<eos::models::BarotropicClosureEOS EOST>
template<typename OperandT>
int serif::equations::BarotropicClosure< EOST >::additional_order (
OperandT ,
int) const [inline], [nodiscard]

10.20.3.2 evaluate()

template<eos::models::BarotropicClosureEOS EOST>
PointLinearization< Operands > serif::equations::BarotropicClosure< EOST >::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

10.20.4 Member Data Documentation

10.20.4.1 m_enthalpy_power

template<eos::models::BarotropicClosureEOS EOST>
double serif::equations::BarotropicClosure< EOST >::m_enthalpy_power [private]

10.20.4.2 m_eos

template<eos::models::BarotropicClosureEOS EOST>
EOST serif::equations::BarotropicClosure< EOST >::m_eos [private]

10.20.4.3 name

template<eos::models::BarotropicClosureEOS EOST>
std::string_view serif::equations::BarotropicClosure< EOST >::name = "barotropic closure" [static], [constexpr]

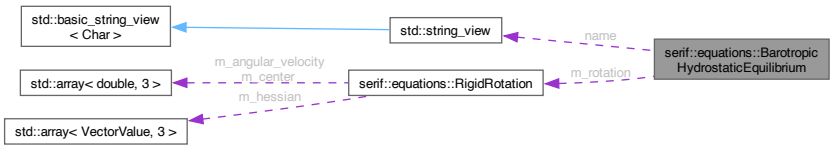
The documentation for this class was generated from the following file:

- src/include/serif/equations/barotropic_closure.hpp

10.21 serif::equations::BarotropicHydrostaticEquilibrium Class Reference

#include <hydrostatic_equilibrium.hpp>

Collaboration diagram for serif::equations::BarotropicHydrostaticEquilibrium:



Public Types

- using Test = discretization::blocks::enthalpy::specific
- using Potential = discretization::blocks::gravity::potential
- using Constant = discretization::blocks::barotropic_constant::mass_normalization
- using AngularVelocity = discretization::blocks::rotation::angular_velocity
- using Operands

Public Member Functions

- BarotropicHydrostaticEquilibrium (RigidRotation::Vector axis={0.0, 0.0, 1.0}, RigidRotation::Vector center={})

  Prescribe only the axis and center; C and angular speed are global scalar operands.
- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const

  Evaluate h + Phi - Psi - C and its partial derivatives at a physical point.
- template<typename OperandT>
  int additional_order (OperandT, int position_order) const

  The rotational potential is quadratic in physical position.

Static Public Attributes

- static constexpr std::string_view name = "hydrostatic equilibrium"

Private Attributes

- RigidRotation m_rotation

10.21.1 Member Typedef Documentation

10.21.1.1 AngularVelocity

using serif::equations::BarotropicHydrostaticEquilibrium::AngularVelocity = discretization::blocks::rotation::angular_velocity

10.21.1.2 Constant

using serif::equations::BarotropicHydrostaticEquilibrium::Constant = discretization::blocks::barotropic_constant::mass_normalization

10.21.1.3 Operands

using serif::equations::BarotropicHydrostaticEquilibrium::Operands

Initial value:

```
utils::types::TypeList<discretization::forms::Operand<Test>, discretization::forms::Operand<Potential>,
                       discretization::forms::Operand<Constant>, discretization::forms::Operand<AngularVelocity>>
```

10.21.1.4 Potential

using serif::equations::BarotropicHydrostaticEquilibrium::Potential = discretization::blocks::gravity::potential

10.21.1.5 Test

using serif::equations::BarotropicHydrostaticEquilibrium::Test = discretization::blocks::enthalpy::specific

10.21.2 Constructor & Destructor Documentation

10.21.2.1 BarotropicHydrostaticEquilibrium()

```
serif::equations::BarotropicHydrostaticEquilibrium::BarotropicHydrostaticEquilibrium (
        RigidRotation::Vector axis = {0.0, 0.0, 1.0},
        RigidRotation::Vector center = {})   [inline], [explicit]
```

Prescribe only the axis and center; C and angular speed are global scalar operands.

10.21.3 Member Function Documentation

10.21.3.1 additional_order()

template<typename OperandT>
int serif::equations::BarotropicHydrostaticEquilibrium::additional_order (
OperandT ,
int position_order) const [inline], [nodiscard]

The rotational potential is quadratic in physical position.

10.21.3.2 evaluate()

PointLinearization< Operands > serif::equations::BarotropicHydrostaticEquilibrium::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

Evaluate h + Phi - Psi - C and its partial derivatives at a physical point.

10.21.4 Member Data Documentation

10.21.4.1 m_rotation

RigidRotation serif::equations::BarotropicHydrostaticEquilibrium::m_rotation [private]

10.21.4.2 name

std::string_view serif::equations::BarotropicHydrostaticEquilibrium::name = "hydrostatic equilibrium" [static], [constexpr]

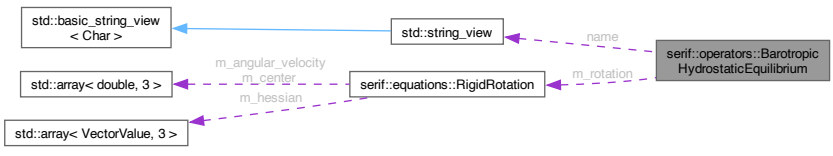
The documentation for this class was generated from the following file:

- src/include/serif/equations/hydrostatic_equilibrium.hpp

10.22 serif::operators::BarotropicHydrostaticEquilibrium Class Reference

#include <hydrostatic_equilibrium.hpp>

Collaboration diagram for serif::operators::BarotropicHydrostaticEquilibrium:



Public Types

- using Test = discretization::blocks::enthalpy::specific
- using Potential = discretization::blocks::gravity::potential
- using Constant = discretization::blocks::barotropic_constant::mass_normalization
- using AngularVelocity = discretization::blocks::rotation::angular_velocity
- using Operands

Public Member Functions

- BarotropicHydrostaticEquilibrium (RigidRotation::Vector axis={0.0, 0.0, 1.0}, RigidRotation::Vector center={})

  Prescribe only the axis and center; C and angular speed are global scalar operands.
- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const

  Evaluate h + Phi - Psi - C and its partial derivatives at a physical point.
- template<typename OperandT>
  int additional_order (OperandT, int position_order) const

  The rotational potential is quadratic in physical position.

Static Public Attributes

- static constexpr std::string_view name = "hydrostatic equilibrium"

Private Attributes

- RigidRotation m_rotation

10.22.1 Member Typedef Documentation

10.22.1.1 AngularVelocity

using serif::equations::BarotropicHydrostaticEquilibrium::AngularVelocity = discretization::blocks::rotation::angular_velocity

10.22.1.2 Constant

using serif::equations::BarotropicHydrostaticEquilibrium::Constant = discretization::blocks::barotropic_constant::mass_normalization

10.22.1.3 Operands

using serif::equations::BarotropicHydrostaticEquilibrium::Operands

Initial value:

```
utils::types::TypeList<discretization::forms::Operand<Test>, discretization::forms::Operand<Potential>,
                       discretization::forms::Operand<Constant>, discretization::forms::Operand<AngularVelocity>>
```

10.22.1.4 Potential

using serif::equations::BarotropicHydrostaticEquilibrium::Potential = discretization::blocks::gravity::potential

10.22.1.5 Test

using serif::equations::BarotropicHydrostaticEquilibrium::Test = discretization::blocks::enthalpy::specific

10.22.2 Constructor & Destructor Documentation

10.22.2.1 BarotropicHydrostaticEquilibrium()

serif::equations::BarotropicHydrostaticEquilibrium::BarotropicHydrostaticEquilibrium (
RigidRotation::Vector axis = {0.0, 0.0, 1.0},
RigidRotation::Vector center = {}) [inline], [explicit]

Prescribe only the axis and center; C and angular speed are global scalar operands.

10.22.3 Member Function Documentation

10.22.3.1 additional_order()

template<typename OperandT>
int serif::equations::BarotropicHydrostaticEquilibrium::additional_order (
OperandT ,
int position_order) const [inline], [nodiscard]

The rotational potential is quadratic in physical position.

10.22.3.2 evaluate()

PointLinearization< Operands > serif::equations::BarotropicHydrostaticEquilibrium::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

Evaluate h + Phi - Psi - C and its partial derivatives at a physical point.

10.22.4 Member Data Documentation

10.22.4.1 m_rotation

RigidRotation serif::equations::BarotropicHydrostaticEquilibrium::m_rotation [private]

10.22.4.2 name

std::string_view serif::equations::BarotropicHydrostaticEquilibrium::name = "hydrostatic equilibrium" [static], [constexpr]

The documentation for this class was generated from the following file:

- src/include/serif/equations/hydrostatic_equilibrium.hpp

10.23 serif::operators::backend::mfem::volume_keys::Basis< Contribution > Struct Template Reference

#include <volume_data.hpp>

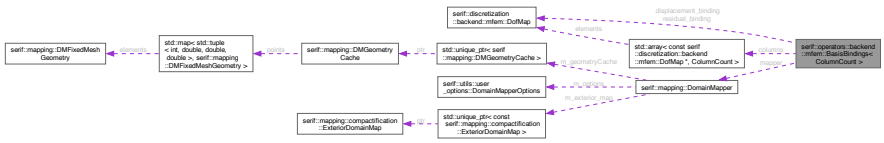
The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.24 serif::operators::backend::mfem::BasisBindings< ColumnCount > Struct Template Reference

#include <volume_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::BasisBindings< ColumnCount >:



Public Member Functions

- bool operator== (const BasisBindings &) const =default

  Basis data changes when either the reference mesh or any coefficient binding changes.

Public Attributes

- const discretization::backend::mfem::DofMap * residual_binding
- const discretization::backend::mfem::DofMap * displacement_binding
- const ::mfem::FiniteElementSpace * compactification_space
- const mapping::DomainMapper * mapper
- long nodes_sequence
- std::array< const discretization::backend::mfem::DofMap *, ColumnCount > columns

10.24.1 Member Function Documentation

10.24.1.1 operator==()

```
template<std::size_t ColumnCount>
bool serif::operators::backend::mfem::BasisBindings< ColumnCount >::operator== (
            const BasisBindings< ColumnCount > & ) const    [default]
```

Basis data changes when either the reference mesh or any coefficient binding changes.

10.24.2 Member Data Documentation

10.24.2.1 columns

```
template<std::size_t ColumnCount>
std::array<const discretization::backend::mfem::DofMap*, ColumnCount> serif::operators::backend::mfem::BasisBindings<
ColumnCount >::columns
```

10.24.2.2 compactification_space

```
template<std::size_t ColumnCount>
const ::mfem::FiniteElementSpace* serif::operators::backend::mfem::BasisBindings< ColumnCount >::compactification_space
```

10.24.2.3 displacement_binding

```
template<std::size_t ColumnCount>
const discretization::backend::mfem::DofMap* serif::operators::backend::mfem::BasisBindings< ColumnCount >::displacement↩
_binding
```

10.24.2.4 mapper

```
template<std::size_t ColumnCount>
const mapping::DomainMapper* serif::operators::backend::mfem::BasisBindings< ColumnCount >::mapper
```

10.24.2.5 nodes_sequence

```
template<std::size_t ColumnCount>
long serif::operators::backend::mfem::BasisBindings< ColumnCount >::nodes_sequence
```

10.24.2.6 residual_binding

template<std::size_t ColumnCount>
const discretization::backend::mfem::DofMap* serif::operators::backend::mfem::BasisBindings< ColumnCount >::residual_↩
binding

The documentation for this struct was generated from the following file:

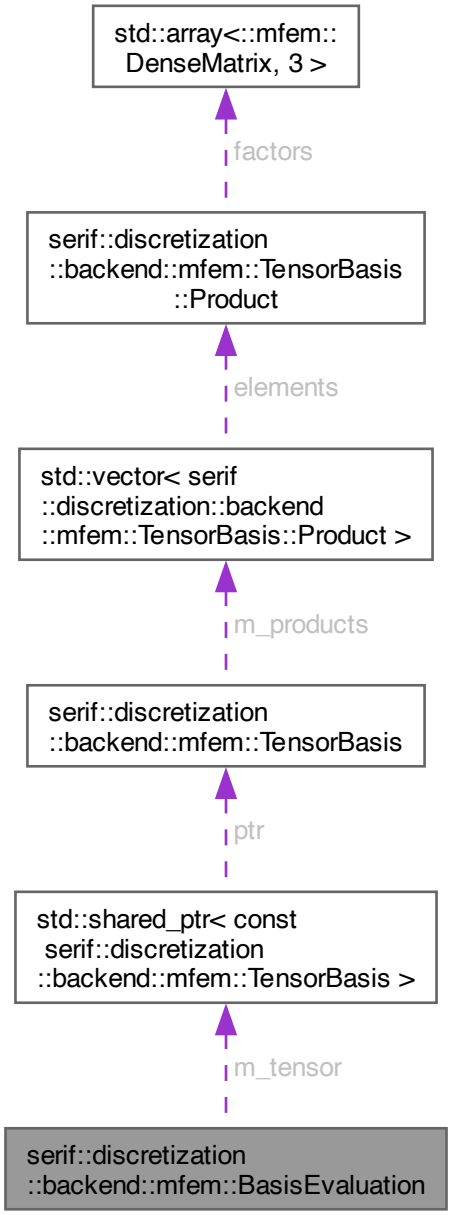
- src/include/serif/operators/backend/mfem/volume_data.hpp

10.25 serif::discretization::backend::mfem::BasisEvaluation Class Reference

Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application.

#include <basis_evaluation.hpp>

Collaboration diagram for serif::discretization::backend::mfem::BasisEvaluation:



Public Member Functions

- void sample (const ::mfem::FiniteElement &element, const ::mfem::IntegrationRule &rule, ReferenceOperation operation, int copies, TensorBasisCache &cache, ContractionMode mode)

Prepare basis values or derivatives at quadrature points using a dense matrix or tensor-product representation.

- void set_constant (int points)

  Set the basis evaluation to a constant basis with the given number of points.
- void reset_from (const BasisEvaluation &reference, int components)

  Allocate storage for a mapped basis using the reference's coefficient ordering and quadrature points.
- template<typename MapT>
  void map_point (int point, const BasisEvaluation &reference, MapT &&map)

  Apply a pointwise linear map to the reference basis or its geometry variation.
- void evaluate (const ::mfem::Vector &coefficients, ::mfem::Vector &values, BasisWorkspace &workspace) const

  Apply the basis to native coefficients; no integration weight is introduced.
- void integrate (const ::mfem::Vector &values, ::mfem::Vector &coefficients, BasisWorkspace &workspace) const

  Apply the coordinate transpose to already weighted quadrature duals.
- int height () const noexcept

  Number of flattened quadrature samples.
- int width () const noexcept

  Number of native element coefficients.
- bool uses_tensor () const noexcept

  Report whether contractions use verified Cartesian factors.

Private Attributes

- int m_points {0}

  Number of quadrature points in the rule.
- int m_components {0}

  Number of output components per point.
- int m_width {0}

  Number of native element coefficients.
- std::shared_ptr< const TensorBasis > m_tensor

  Canonical tensor representation, when the tensor path is selected.
- ::mfem::DenseMatrix m_dense

  Full evaluation matrix, including applied mappings, when the dense path is selected.
- ::mfem::DenseMatrix m_mapping

  Stacked pointwise maps from canonical tensor samples to current output components.

10.25.1 Detailed Description

Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application.

Consider a scalar field represented on one finite element:

$$\rho_h(\xi) = \sum_{j=1}^{n} c_j \phi_j(\xi).$$

Here, c_j is a scalar coefficient, ϕ_j is its associated basis function, and ξ is the canonical reference coordinate. The field is expressed in these coordinates through the element mapping.

For a quadrature rule with N_q points, define the evaluation matrix $\mathbf{E}$ by

$$E_{qj} = \phi_j(\xi_q).$$

$\mathbf{E}$ therefore has N_q rows and n columns. The field value at one point is

$$\rho_h(\xi_q) = \sum_{j=1}^{n} E_{qj} c_j.$$

Collecting the samples into a vector gives

$$\rho_h = \mathbf{E}\mathbf{c}.$$

This class represents that linear operation without necessarily storing the full matrix $\mathbf{E}$: it can instead apply equivalent tensor contractions.

Example

Evaluate a constant scalar field on the reference cube, then compute its integral against each basis function.

```
namespace backend = serif::discretization::backend::mfem;

// First-order nodal H1 basis on the reference cube.
::mfem::H1_HexahedronElement element(1);
const ::mfem::IntegrationRule& rule =
    ::mfem::IntRules.Get(::mfem::Geometry::CUBE, 2);

backend::TensorBasisCache cache;
backend::BasisWorkspace workspace;
backend::BasisEvaluation basis;

// Prepare E: one scalar basis value per quadrature point and DOF.
basis.sample(
    element,
    rule,
    backend::ReferenceOperation::value,
    1, // One copy of the scalar basis.
    cache,
    backend::ContractionMode::automatic
);

// For this nodal basis, equal coefficients represent a constant field.
::mfem::Vector coefficients(element.GetDof());
coefficients = 2.0;

// Compute values = E * coefficients.
// Every quadrature-point value is approximately 2.0.
::mfem::Vector values;
basis.evaluate(coefficients, values, workspace);

// Supply quadrature weights explicitly: integrate does not add them.
::mfem::Vector weighted_values(rule.GetNPoints());
for (int q = 0; q < rule.GetNPoints(); ++q) {
    weighted_values[q] = rule.IntPoint(q).weight * values[q];
}

// Compute element_integrals = E^T * weighted_values.
::mfem::Vector element_integrals;
basis.integrate(weighted_values, element_integrals, workspace);
```

Each output entry approximates

$$\int_{\hat{K}} \phi_j(\xi)\rho_h(\xi)\, d\xi.$$

These are element-DOF integral contributions, not reconstructed field coefficients. Integration over a physical element additionally requires the appropriate volume Jacobian in the quadrature weights.

10.25.2 Member Function Documentation

10.25.2.1 evaluate()

void serif::discretization::backend::mfem::BasisEvaluation::evaluate (
const ::mfem::Vector & coefficients,
::mfem::Vector & values,
BasisWorkspace & workspace) const

Apply the basis to native coefficients; no integration weight is introduced.

Parameters

coefficients	Input coefficients to apply the basis to.
values	Output values after applying the basis.
workspace	Workspace to use for intermediate computations.

10.25.2.2 height()

int serif::discretization::backend::mfem::BasisEvaluation::height () const [nodiscard], [noexcept]

Number of flattened quadrature samples.

10.25.2.3 integrate()

void serif::discretization::backend::mfem::BasisEvaluation::integrate (
const ::mfem::Vector & values,
::mfem::Vector & coefficients,
BasisWorkspace & workspace) const

Apply the coordinate transpose to already weighted quadrature duals.

Parameters

in	values	Quadrature data with any required integration weights already applied.
out	coefficients	Element-DOF contributions computed as E^T times values.
	workspace	Workspace to use for intermediate computations.

10.25.2.4 map_point()

template<typename MapT>
void serif::discretization::backend::mfem::BasisEvaluation::map_point (
int point,
const BasisEvaluation & reference,
MapT && map)

Apply a pointwise linear map to the reference basis or its geometry variation.

Template Parameters

MapT	Type of the mapping function to apply.

Parameters

point	Quadrature point index to apply the mapping at.
reference	Reference basis evaluation to map from.
map	Mapping function to apply.

Precondition

The mapping callback must be linear in its input and fill every output component.

10.25.2.5 reset_from()

```
void serif::discretization::backend::mfem::BasisEvaluation::reset_from (
        const BasisEvaluation & reference,
        int components)
```

Allocate storage for a mapped basis using the reference's coefficient ordering and quadrature points.

Parameters

reference	Basis evaluation supplying the dimensions and any shared tensor representation.
components	Number of output components per quadrature point.

Note

Populate the destination at every quadrature point using map_point before evaluation or transpose application.

10.25.2.6 sample()

```
void serif::discretization::backend::mfem::BasisEvaluation::sample (
        const ::mfem::FiniteElement & element,
        const ::mfem::IntegrationRule & rule,
        ReferenceOperation operation,
        int copies,
        TensorBasisCache & cache,
        ContractionMode mode)
```

Prepare basis values or derivatives at quadrature points using a dense matrix or tensor-product representation.

Parameters

element	Finite element to sample.
rule	Integration rule to use for sampling.
operation	Reference operation to perform.
copies	Number of independent copies of the finite-element basis; for example, three scalar basis copies for a vector H1 field.
cache	Tensor basis cache to use.
mode	Contraction mode to use.

10.25.2.7 set_constant()

void serif::discretization::backend::mfem::BasisEvaluation::set_constant (
int points)

Set the basis evaluation to a constant basis with the given number of points.

Parameters

points	Number of quadrature points to use for the constant basis.

10.25.2.8 uses_tensor()

bool serif::discretization::backend::mfem::BasisEvaluation::uses_tensor () const [nodiscard], [noexcept]

Report whether contractions use verified Cartesian factors.

10.25.2.9 width()

int serif::discretization::backend::mfem::BasisEvaluation::width () const [nodiscard], [noexcept]

Number of native element coefficients.

10.25.3 Member Data Documentation

10.25.3.1 m_components

int serif::discretization::backend::mfem::BasisEvaluation::m_components {0} [private]

Number of output components per point.

10.25.3.2 m_dense

::mfem::DenseMatrix serif::discretization::backend::mfem::BasisEvaluation::m_dense [private]

Full evaluation matrix, including applied mappings, when the dense path is selected.

10.25.3.3 m_mapping

::mfem::DenseMatrix serif::discretization::backend::mfem::BasisEvaluation::m_mapping [private]

Stacked pointwise maps from canonical tensor samples to current output components.

10.25.3.4 m_points

int serif::discretization::backend::mfem::BasisEvaluation::m_points {0} [private]

Number of quadrature points in the rule.

10.25.3.5 m_tensor

std::shared_ptr<const TensorBasis> serif::discretization::backend::mfem::BasisEvaluation::m_tensor [private]

Canonical tensor representation, when the tensor path is selected.

10.25.3.6 m_width

int serif::discretization::backend::mfem::BasisEvaluation::m_width {0} [private]

Number of native element coefficients.

The documentation for this class was generated from the following files:

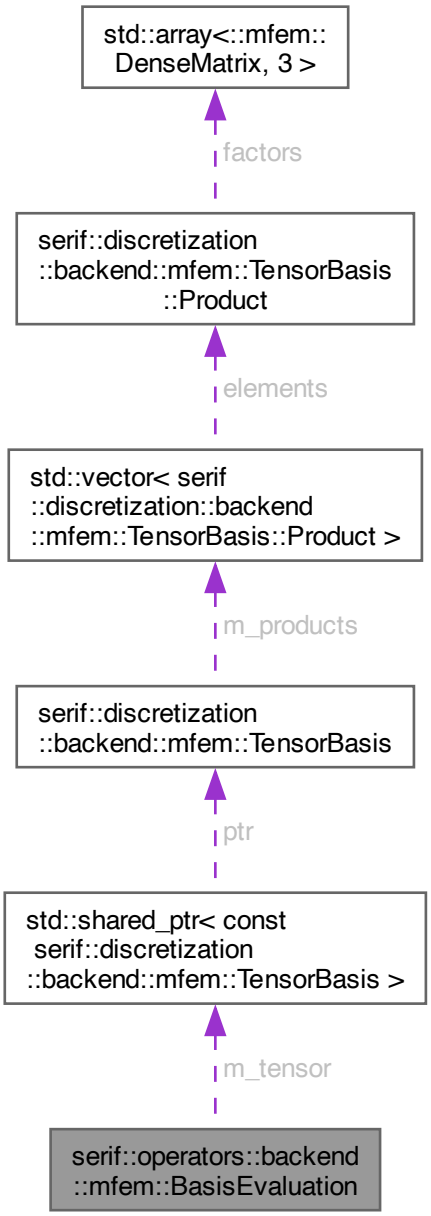
- src/include/serif/discretization/backend/mfem/basis_evaluation.hpp
- src/lib/serif/discretization/backend/mfem/basis_evaluation.cpp

10.26 serif::operators::backend::mfem::BasisEvaluation Class Reference

Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application.

#include <basis_evaluation.hpp>

Collaboration diagram for serif::operators::backend::mfem::BasisEvaluation:



Public Member Functions

- void sample (const ::mfem::FiniteElement &element, const ::mfem::IntegrationRule &rule, ReferenceOperation operation, int copies, TensorBasisCache &cache, ContractionMode mode)

  Prepare basis values or derivatives at quadrature points using a dense matrix or tensor-product representation.
- void set_constant (int points)

  Set the basis evaluation to a constant basis with the given number of points.
- void reset_from (const BasisEvaluation &reference, int components)

  Allocate storage for a mapped basis using the reference's coefficient ordering and quadrature points.
- template<typename MapT>
  void map_point (int point, const BasisEvaluation &reference, MapT &&map)

  Apply a pointwise linear map to the reference basis or its geometry variation.
- void evaluate (const ::mfem::Vector &coefficients, ::mfem::Vector &values, BasisWorkspace &workspace) const

  Apply the basis to native coefficients; no integration weight is introduced.
- void integrate (const ::mfem::Vector &values, ::mfem::Vector &coefficients, BasisWorkspace &workspace) const

  Apply the coordinate transpose to already weighted quadrature duals.
- int height () const noexcept

  Number of flattened quadrature samples.
- int width () const noexcept

  Number of native element coefficients.
- bool uses_tensor () const noexcept

  Report whether contractions use verified Cartesian factors.

Private Attributes

- int m_points {0}

  Number of quadrature points in the rule.
- int m_components {0}

  Number of output components per point.
- int m_width {0}

  Number of native element coefficients.
- std::shared_ptr< const TensorBasis > m_tensor

  Canonical tensor representation, when the tensor path is selected.
- ::mfem::DenseMatrix m_dense

  Full evaluation matrix, including applied mappings, when the dense path is selected.
- ::mfem::DenseMatrix m_mapping

  Stacked pointwise maps from canonical tensor samples to current output components.

10.26.1 Detailed Description

Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application.

Consider a scalar field represented on one finite element:

$$\rho_h(\xi) = \sum_{j=1}^{n} c_j \phi_j(\xi).$$

Here, c_j is a scalar coefficient, ϕ_j is its associated basis function, and ξ is the canonical reference coordinate. The field is expressed in these coordinates through the element mapping.

For a quadrature rule with N_q points, define the evaluation matrix $\mathbf{E}$ by

$$E_{qj} = \phi_j(\xi_q).$$

$\mathbf{E}$ therefore has N_q rows and n columns. The field value at one point is

$$\rho_h(\xi_q) = \sum_{j=1}^{n} E_{qj} c_j.$$

Collecting the samples into a vector gives

$$\rho_h = \mathbf{E}\mathbf{c}.$$

This class represents that linear operation without necessarily storing the full matrix $\mathbf{E}$: it can instead apply equivalent tensor contractions.

Example

Evaluate a constant scalar field on the reference cube, then compute its integral against each basis function.

```
namespace backend = serif::discretization::backend::mfem;

// First-order nodal H1 basis on the reference cube.
::mfem::H1_HexahedronElement element(1);
const ::mfem::IntegrationRule& rule =
    ::mfem::IntRules.Get(::mfem::Geometry::CUBE, 2);

backend::TensorBasisCache cache;
backend::BasisWorkspace workspace;
backend::BasisEvaluation basis;

// Prepare E: one scalar basis value per quadrature point and DOF.
basis.sample(
    element,
    rule,
    backend::ReferenceOperation::value,
    1, // One copy of the scalar basis.
    cache,
    backend::ContractionMode::automatic
```

```
);

// For this nodal basis, equal coefficients represent a constant field.
::mfem::Vector coefficients(element.GetDof());
coefficients = 2.0;

// Compute values = E * coefficients.
// Every quadrature-point value is approximately 2.0.
::mfem::Vector values;
basis.evaluate(coefficients, values, workspace);

// Supply quadrature weights explicitly: integrate does not add them.
::mfem::Vector weighted_values(rule.GetNPoints());
for (int q = 0; q < rule.GetNPoints(); ++q) {
    weighted_values[q] = rule.IntPoint(q).weight * values[q];
}

// Compute element_integrals = E^T * weighted_values.
::mfem::Vector element_integrals;
basis.integrate(weighted_values, element_integrals, workspace);
```

Each output entry approximates

$$\int_{\hat{K}} \phi_j(\xi)\rho_h(\xi)\, d\xi.$$

These are element-DOF integral contributions, not reconstructed field coefficients. Integration over a physical element additionally requires the appropriate volume Jacobian in the quadrature weights.

10.26.2 Member Function Documentation

10.26.2.1 evaluate()

void serif::discretization::backend::mfem::BasisEvaluation::evaluate (
const ::mfem::Vector & coefficients,
::mfem::Vector & values,
BasisWorkspace & workspace) const

Apply the basis to native coefficients; no integration weight is introduced.

Parameters

coefficients	Input coefficients to apply the basis to.
values	Output values after applying the basis.
workspace	Workspace to use for intermediate computations.

10.26.2.2 height()

int serif::discretization::backend::mfem::BasisEvaluation::height () const [nodiscard], [noexcept]

Number of flattened quadrature samples.

10.26.2.3 integrate()

void serif::discretization::backend::mfem::BasisEvaluation::integrate (
const ::mfem::Vector & values,
::mfem::Vector & coefficients,
BasisWorkspace & workspace) const

Apply the coordinate transpose to already weighted quadrature duals.

Parameters

in	values	Quadrature data with any required integration weights already applied.
out	coefficients	Element-DOF contributions computed as E^T times values.
	workspace	Workspace to use for intermediate computations.

10.26.2.4 map_point()

```
template<typename MapT>
void serif::discretization::backend::mfem::BasisEvaluation::map_point (
                int point,
                const BasisEvaluation & reference,
                MapT && map)
```

Apply a pointwise linear map to the reference basis or its geometry variation.

Template Parameters

MapT	Type of the mapping function to apply.

Parameters

point	Quadrature point index to apply the mapping at.
reference	Reference basis evaluation to map from.
map	Mapping function to apply.

Precondition

The mapping callback must be linear in its input and fill every output component.

10.26.2.5 reset_from()

```
void serif::discretization::backend::mfem::BasisEvaluation::reset_from (
                const BasisEvaluation & reference,
                int components)
```

Allocate storage for a mapped basis using the reference's coefficient ordering and quadrature points.

Parameters

reference	Basis evaluation supplying the dimensions and any shared tensor representation.
components	Number of output components per quadrature point.

Note

Populate the destination at every quadrature point using map_point before evaluation or transpose application.

10.26.2.6 sample()

void serif::discretization::backend::mfem::BasisEvaluation::sample (
const ::mfem::FiniteElement & element,
const ::mfem::IntegrationRule & rule,
ReferenceOperation operation,
int copies,
TensorBasisCache & cache,
ContractionMode mode)

Prepare basis values or derivatives at quadrature points using a dense matrix or tensor-product representation.

Parameters

element	Finite element to sample.
rule	Integration rule to use for sampling.
operation	Reference operation to perform.
copies	Number of independent copies of the finite-element basis; for example, three scalar basis copies for a vector H1 field.
cache	Tensor basis cache to use.
mode	Contraction mode to use.

10.26.2.7 set_constant()

void serif::discretization::backend::mfem::BasisEvaluation::set_constant (
int points)

Set the basis evaluation to a constant basis with the given number of points.

Parameters

points	Number of quadrature points to use for the constant basis.

10.26.2.8 uses_tensor()

bool serif::discretization::backend::mfem::BasisEvaluation::uses_tensor () const [nodiscard], [noexcept]

Report whether contractions use verified Cartesian factors.

10.26.2.9 width()

int serif::discretization::backend::mfem::BasisEvaluation::width () const [nodiscard], [noexcept]

Number of native element coefficients.

10.26.3 Member Data Documentation

10.26.3.1 m_components

int serif::discretization::backend::mfem::BasisEvaluation::m_components {0} [private]

Number of output components per point.

10.26.3.2 m_dense

::mfem::DenseMatrix serif::discretization::backend::mfem::BasisEvaluation::m_dense [private]

Full evaluation matrix, including applied mappings, when the dense path is selected.

10.26.3.3 m_mapping

::mfem::DenseMatrix serif::discretization::backend::mfem::BasisEvaluation::m_mapping [private]

Stacked pointwise maps from canonical tensor samples to current output components.

10.26.3.4 m_points

int serif::discretization::backend::mfem::BasisEvaluation::m_points {0} [private]

Number of quadrature points in the rule.

10.26.3.5 m_tensor

std::shared_ptr<const TensorBasis> serif::discretization::backend::mfem::BasisEvaluation::m_tensor [private]

Canonical tensor representation, when the tensor path is selected.

10.26.3.6 m_width

int serif::discretization::backend::mfem::BasisEvaluation::m_width {0} [private]

Number of native element coefficients.

The documentation for this class was generated from the following files:

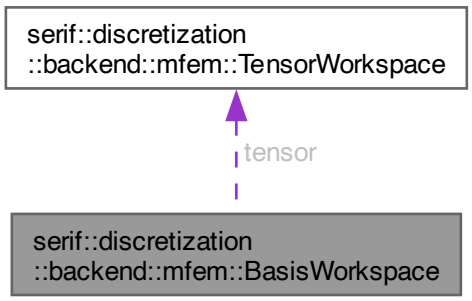
- src/include/serif/discretization/backend/mfem/basis_evaluation.hpp
- src/lib/serif/discretization/backend/mfem/basis_evaluation.cpp

10.27 serif::discretization::backend::mfem::BasisWorkspace Struct Reference

Reusable scratch storage for basis contractions, reducing repeated allocations.

#include <basis_evaluation.hpp>

Collaboration diagram for serif::discretization::backend::mfem::BasisWorkspace:



Public Attributes

- TensorWorkspace tensor
- ::mfem::Vector reference_values

10.27.1 Detailed Description

Reusable scratch storage for basis contractions, reducing repeated allocations.

10.27.2 Member Data Documentation

10.27.2.1 reference_values

::mfem::Vector serif::discretization::backend::mfem::BasisWorkspace::reference_values

10.27.2.2 tensor

TensorWorkspace serif::discretization::backend::mfem::BasisWorkspace::tensor

The documentation for this struct was generated from the following file:

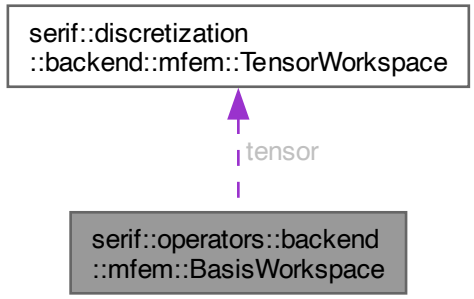
- src/include/serif/discretization/backend/mfem/basis_evaluation.hpp

10.28 serif::operators::backend::mfem::BasisWorkspace Struct Reference

Reusable scratch storage for basis contractions, reducing repeated allocations.

#include <basis_evaluation.hpp>

Collaboration diagram for serif::operators::backend::mfem::BasisWorkspace:



Public Attributes

- TensorWorkspace tensor
- ::mfem::Vector reference_values

10.28.1 Detailed Description

Reusable scratch storage for basis contractions, reducing repeated allocations.

10.28.2 Member Data Documentation

10.28.2.1 reference_values

::mfem::Vector serif::discretization::backend::mfem::BasisWorkspace::reference_values

10.28.2.2 tensor

TensorWorkspace serif::discretization::backend::mfem::BasisWorkspace::tensor

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/backend/mfem/basis_evaluation.hpp

10.29 serif::utils::dependencies::Binding< KeyT, TargetT > Struct Template Reference

#include <binding.hpp>

Public Types

- using Key = KeyT
- using Target = TargetT

10.29.1 Member Typedef Documentation

10.29.1.1 Key

template<typename KeyT, typename TargetT>
using serif::utils::dependencies::Binding< KeyT, TargetT >::Key = KeyT

10.29.1.2 Target

template<typename KeyT, typename TargetT>
using serif::utils::dependencies::Binding< KeyT, TargetT >::Target = TargetT

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/binding.hpp

10.30 serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< K, T > Struct Template Reference

Public Types

- using type = T

10.30.1 Member Typedef Documentation

10.30.1.1 type

template<typename KeyT, IsBinding... BindingTs>
template<typename K, typename T>
using serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< K, T >::type = T

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/binding.hpp

10.31 serif::utils::dependencies::Bindings< BindingTs > Struct Template Reference

#include <binding.hpp>

The documentation for this struct was generated from the following file:

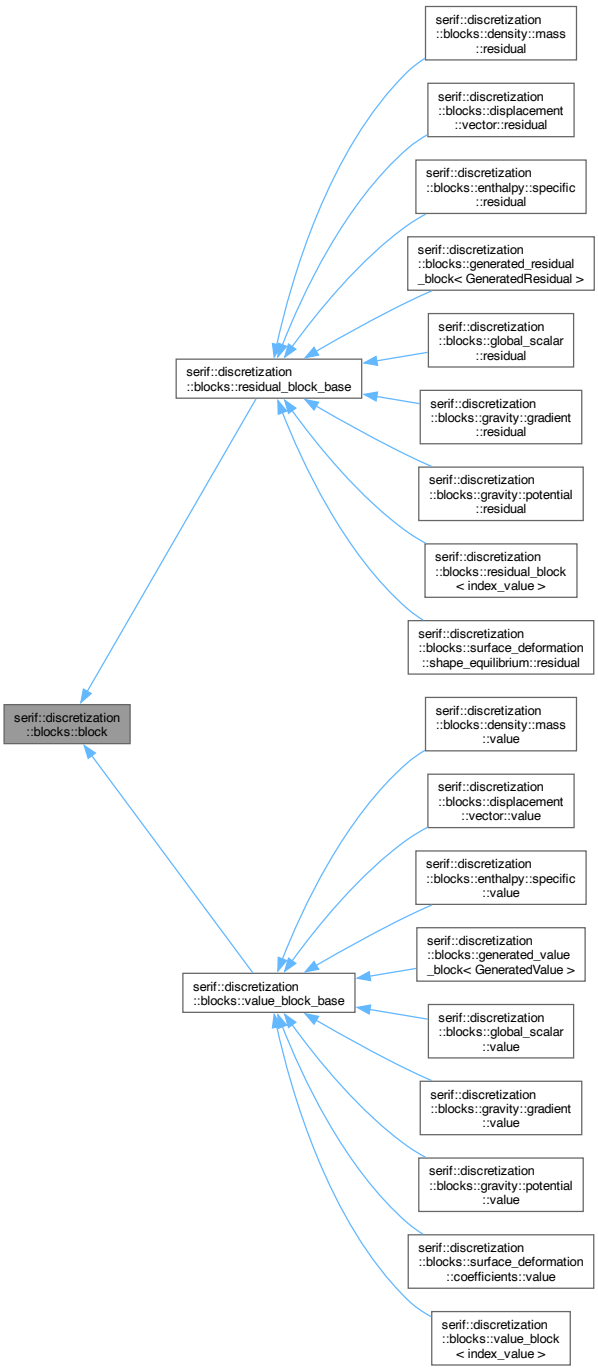
- src/include/serif/utils/dependencies/binding.hpp

10.32 serif::discretization::blocks::block Struct Reference

Base class for all block types in the discretization framework.

#include <base.hpp>

Inheritance diagram for serif::discretization::blocks::block:



10.32.1 Detailed Description

Base class for all block types in the discretization framework.

The documentation for this struct was generated from the following file:

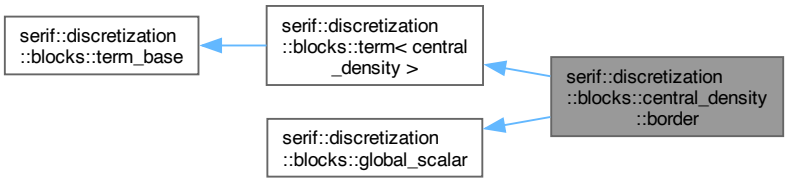
- src/include/serif/discretization/blocks/base.hpp

10.33 serif::discretization::blocks::central_density::border Struct Reference

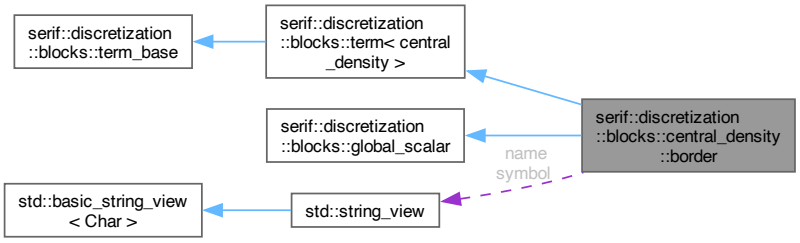
The border term represents the border row used to enforce a central density constraint. This is a global scalar as it is a single value over the entire domain.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::central_density::border:



Collaboration diagram for serif::discretization::blocks::central_density::border:



Public Types

- using Support = domain::StellarDomains

  We only enforce this over the stellar domain as there is only material to "slosh" around in the stellar domain.
- using Quantity = dimensions::SpecificEnthalpy

  As we are in the barotropic case this border term is associate with the specific enthalpy.

Public Types inherited from serif::discretization::blocks::term< central_density >

- using field_type

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<border>()
  Compile time reflected name of the term.
- static constexpr std::string_view symbol = " _c"
  Compile time symbol used for the term.

10.33.1 Detailed Description

The border term represents the border row used to enforce a central density constraint. This is a global scalar as it is a single value over the entire domain.

10.33.2 Member Typedef Documentation

10.33.2.1 Quantity

using serif::discretization::blocks::central_density::border::Quantity = dimensions::SpecificEnthalpy

As we are in the barotropic case this border term is associate with the specific enthalpy.

10.33.2.2 Support

using serif::discretization::blocks::central_density::border::Support = domain::StellarDomains

We only enforce this over the stellar domain as there is only material to "slosh" around in the stellar domain.

10.33.3 Member Data Documentation

10.33.3.1 name

std::string_view serif::discretization::blocks::central_density::border::name = meta::type_name<border>() [static], [constexpr]

Compile time reflected name of the term.

10.33.3.2 symbol

std::string_view serif::discretization::blocks::central_density::border::symbol = "_c" [static], [constexpr]

Compile time symbol used for the term.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

10.34 serif::operators::backend::mfem::nodal_keys::BorderLinearization Struct Reference

#include <nodal_inputs.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/nodal_inputs.hpp

10.35 serif::utils::dependencies::detail::BorrowedValue< T > Struct Template Reference

#include <resource.hpp>

Public Types

- using Target = T

Public Member Functions

- const T & get () const

  Borrowing does not extend the lifetime of the source.

Public Attributes

- std::reference_wrapper< const T > value

10.35.1 Member Typedef Documentation

10.35.1.1 Target

template<typename T>
using serif::utils::dependencies::detail::BorrowedValue< T >::Target = T

10.35.2 Member Function Documentation

10.35.2.1 get()

template<typename T>
const T & serif::utils::dependencies::detail::BorrowedValue< T >::get () const [inline], [nodiscard]

Borrowing does not extend the lifetime of the source.

10.35.3 Member Data Documentation

10.35.3.1 value

template<typename T>
std::reference_wrapper<const T> serif::utils::dependencies::detail::BorrowedValue< T >::value

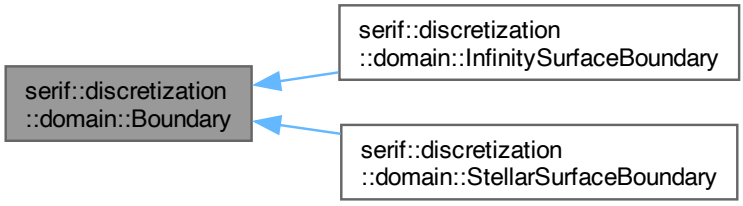
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/resource.hpp

10.36 serif::discretization::domain::Boundary Struct Reference

#include <types.hpp>

Inheritance diagram for serif::discretization::domain::Boundary:



The documentation for this struct was generated from the following file:

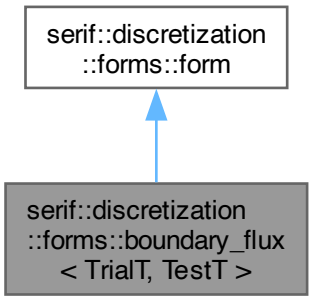
- src/include/serif/discretization/domain/types.hpp

10.37 serif::discretization::forms::boundary_flux< TrialT, TestT > Struct Template Reference

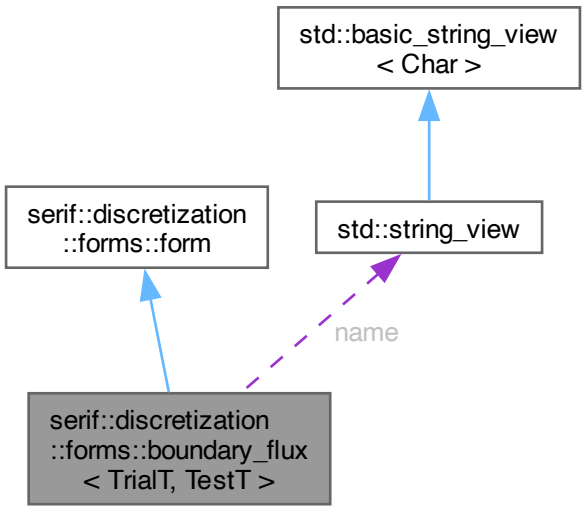
The boundary_flux form represents the boundary flux operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the boundary flux operation.

#include <forms.hpp>

Inheritance diagram for serif::discretization::forms::boundary_flux< TrialT, TestT >:



Collaboration diagram for serif::discretization::forms::boundary_flux< TrialT, TestT >:



Public Types

- using trial_type = TrialT

  The trial term type for the boundary flux form.
- using test_type = TestT

  The test term type for the boundary flux form.
- using operand_types = utils::types::TypeList<Operand<TrialT, Operation::NormalTrace>, Operand<TestT, Operation::NormalTrace>>

  Type list of operand types for the boundary flux form, used for compile-time reflection and iteration over the operands.

Static Public Attributes

- static constexpr std::string_view name = meta::template_name<boundary_flux>()

  Compile time reflected name of the form.

10.37.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::boundary_flux< TrialT, TestT >

The boundary_flux form represents the boundary flux operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the boundary flux operation.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

10.37.2 Member Typedef Documentation

10.37.2.1 operand_types

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::boundary_flux< TrialT, TestT >::operand_types = utils::types::TypeList<Operand<TrialT, Operation::NormalTrace>, Operand<TestT, Operation::NormalTrace>>

Type list of operand types for the boundary flux form, used for compile-time reflection and iteration over the operands.

10.37.2.2 test_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::boundary_flux< TrialT, TestT >::test_type = TestT

The test term type for the boundary flux form.

10.37.2.3 trial_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::boundary_flux< TrialT, TestT >::trial_type = TrialT

The trial term type for the boundary flux form.

10.37.3 Member Data Documentation

10.37.3.1 name

template<blocks::Term TrialT, blocks::Term TestT>
std::string_view serif::discretization::forms::boundary_flux< TrialT, TestT >::name = meta::template_name<boundary_flux>()
[static], [constexpr]

Compile time reflected name of the form.

The documentation for this struct was generated from the following file:

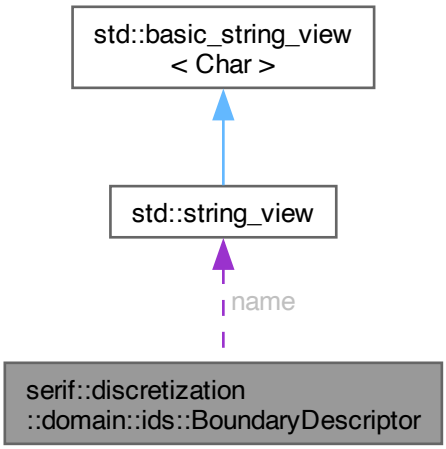
- src/include/serif/discretization/forms/forms.hpp

10.38 serif::discretization::domain::ids::BoundaryDescriptor Struct Reference

The BoundaryDescriptor struct is a runtime representation of a boundary, containing its name and integer attribute ID. This is used to project compile-time BoundaryID types into runtime values.

#include <boundary.hpp>

Collaboration diagram for serif::discretization::domain::ids::BoundaryDescriptor:



Public Attributes

- std::string_view name

  The name of the boundary, used for runtime identification.
- int ID

  The integer attribute value in the mesh file that corresponds to this boundary.

10.38.1 Detailed Description

The BoundaryDescriptor struct is a runtime representation of a boundary, containing its name and integer attribute ID. This is used to project compile-time BoundaryID types into runtime values.

10.38.2 Member Data Documentation

10.38.2.1 ID

int serif::discretization::domain::ids::BoundaryDescriptor::ID

The integer attribute value in the mesh file that corresponds to this boundary.

10.38.2.2 name

std::string_view serif::discretization::domain::ids::BoundaryDescriptor::name

The name of the boundary, used for runtime identification.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/boundary.hpp

10.39 serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue > Struct Template Reference

The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation.

#include <boundary.hpp>

Public Types

- using boundary_type = BoundaryT

  The semantic boundary type associated with this BoundaryID.

Static Public Attributes

- static constexpr int ID = IdentifierValue

  The integer attribute value in the mesh file that corresponds to this boundary.

10.39.1 Detailed Description

template<IsBoundary BoundaryT, int IdentifierValue>
struct serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >

The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation.

Template Parameters

BoundaryT	The semantic boundary type (must satisfy the IsBoundary concept).
IdentifierValue	The integer attribute value in the mesh file that corresponds to this boundary.

10.39.2 Member Typedef Documentation

10.39.2.1 boundary_type

template<IsBoundary BoundaryT, int IdentifierValue>
using serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >::boundary_type = BoundaryT

The semantic boundary type associated with this BoundaryID.

10.39.3 Member Data Documentation

10.39.3.1 ID

template<IsBoundary BoundaryT, int IdentifierValue>
int serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >::ID = IdentifierValue [static], [constexpr]

The integer attribute value in the mesh file that corresponds to this boundary.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/boundary.hpp

10.40 serif::discretization::domain::schema::BoundaryID< BoundaryT, IdentifierValue > Struct Template Reference

The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation.

#include <boundary.hpp>

Public Types

- using boundary_type = BoundaryT

  The semantic boundary type associated with this BoundaryID.

Static Public Attributes

- static constexpr int ID = IdentifierValue

  The integer attribute value in the mesh file that corresponds to this boundary.

10.40.1 Detailed Description

template<IsBoundary BoundaryT, int IdentifierValue>
struct serif::discretization::domain::schema::BoundaryID< BoundaryT, IdentifierValue >

The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation.

Template Parameters

BoundaryT	The semantic boundary type (must satisfy the IsBoundary concept).
IdentifierValue	The integer attribute value in the mesh file that corresponds to this boundary.

10.40.2 Member Typedef Documentation

10.40.2.1 boundary_type

template<IsBoundary BoundaryT, int IdentifierValue>
using serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >::boundary_type = BoundaryT

The semantic boundary type associated with this BoundaryID.

10.40.3 Member Data Documentation

10.40.3.1 ID

template<IsBoundary BoundaryT, int IdentifierValue>
int serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >::ID = IdentifierValue [static], [constexpr]

The integer attribute value in the mesh file that corresponds to this boundary.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/boundary.hpp

10.41 serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs > Struct Template Reference

A list of boundary IDs in the discretization framework.

#include <lists.hpp>

Static Public Member Functions

- static constexpr std::array< BoundaryDescriptor, count > descriptors () noexcept
  Get an array of BoundaryDescriptor objects for the boundary IDs in the list.

Static Public Attributes

- static constexpr std::size_t count = CountVariadicArguments<BoundaryIDTs...>()
  The number of boundary IDs in the list.

10.41.1 Detailed Description

template<IsBoundaryID... BoundaryIDTs>
requires validation::AreDistinctBoundaryIDs<BoundaryIDTs...>
struct serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs >

A list of boundary IDs in the discretization framework.

10.41.2 Member Function Documentation

10.41.2.1 descriptors()

template<IsBoundaryID... BoundaryIDTs>
static constexpr std::array< BoundaryDescriptor, count > serif::discretization::domain::ids::lists::BoundaryIDList< Boundary↩
IDTs >::descriptors () [inline], [static], [nodiscard], [constexpr], [noexcept]

Get an array of BoundaryDescriptor objects for the boundary IDs in the list.

Returns

An array of BoundaryDescriptor objects, each containing the name and ID of a boundary.

10.41.3 Member Data Documentation

10.41.3.1 count

template<IsBoundaryID... BoundaryIDTs>
std::size_t serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs >::count = CountVariadicArguments<Boundary↩
IDTs...>() [static], [constexpr]

The number of boundary IDs in the list.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/lists/lists.hpp

10.42 serif::discretization::domain::schema::BoundaryIDList< BoundaryIDTs > Struct Template Reference

A list of boundary IDs in the discretization framework.

#include <lists.hpp>

Static Public Member Functions

- static constexpr std::array< BoundaryDescriptor, count > descriptors () noexcept
  Get an array of BoundaryDescriptor objects for the boundary IDs in the list.

Static Public Attributes

- static constexpr std::size_t count = CountVariadicArguments<BoundaryIDTs...>()
  The number of boundary IDs in the list.

10.42.1 Detailed Description

template<IsBoundaryID... BoundaryIDTs>
requires validation::AreDistinctBoundaryIDs<BoundaryIDTs...>
struct serif::discretization::domain::schema::BoundaryIDList< BoundaryIDTs >

A list of boundary IDs in the discretization framework.

10.42.2 Member Function Documentation

10.42.2.1 descriptors()

template<IsBoundaryID... BoundaryIDTs>
static constexpr std::array< BoundaryDescriptor, count > serif::discretization::domain::ids::lists::BoundaryIDList< Boundary↩
IDTs >::descriptors () [inline], [static], [nodiscard], [constexpr], [noexcept]

Get an array of BoundaryDescriptor objects for the boundary IDs in the list.

Returns

An array of BoundaryDescriptor objects, each containing the name and ID of a boundary.

10.42.3 Member Data Documentation

10.42.3.1 count

template<IsBoundaryID... BoundaryIDTs>
std::size_t serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs >::count = CountVariadicArguments<Boundary↩
IDTs...>() [static], [constexpr]

The number of boundary IDs in the list.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/lists/lists.hpp

10.43 serif::discretization::domain::BoundaryIDResolver< BoundaryT, BoundaryIDListT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/resolver.hpp

10.44 serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > > Struct Template Reference

#include <resolver.hpp>

Public Types

- template<typename B>
  using ResolverFor = BoundaryIDResolver<B, ids::lists::BoundaryIDList<BoundaryIDTs...>>

Static Public Member Functions

- static constexpr bool contains_id (const int boundaryID) noexcept

Static Public Attributes

- static constexpr bool registered = (ResolverFor<BoundaryTs>::registered && ...)

10.44.1 Member Typedef Documentation

10.44.1.1 ResolverFor

template<IsBoundary... BoundaryTs, ids::IsBoundaryID... BoundaryIDTs>
template<typename B>
using serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > >::ResolverFor = BoundaryIDResolver<B, ids::lists::BoundaryIDList<BoundaryIDTs...>>

10.44.2 Member Function Documentation

10.44.2.1 contains_id()

template<IsBoundary... BoundaryTs, ids::IsBoundaryID... BoundaryIDTs>
static constexpr bool serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > >::contains_id (
const int boundaryID) [inline], [static], [nodiscard], [constexpr], [noexcept]

10.44.3 Member Data Documentation

10.44.3.1 registered

template<IsBoundary... BoundaryTs, ids::IsBoundaryID... BoundaryIDTs>
bool serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > >::registered = (ResolverFor<BoundaryTs>::registered && ...) [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/resolver.hpp

10.45 serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > > Struct Template Reference

#include <resolver.hpp>

Static Public Member Functions

- static constexpr bool contains_id (const int boundaryID) noexcept
- static consteval int id ()

Static Public Attributes

- static constexpr bool registered = (BoundaryIsIdentifiedBy<BoundaryT, BoundaryIDTs> || ...)

10.45.1 Member Function Documentation

10.45.1.1 contains_id()

template<IsBoundary BoundaryT, ids::IsBoundaryID... BoundaryIDTs>
static constexpr bool serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > >::contains_id (
const int boundaryID) [inline], [static], [nodiscard], [constexpr], [noexcept]

10.45.1.2 id()

template<IsBoundary BoundaryT, ids::IsBoundaryID... BoundaryIDTs>
static consteval int serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > >::id () [inline], [static], [nodiscard], [consteval]

10.45.2 Member Data Documentation

10.45.2.1 registered

template<IsBoundary BoundaryT, ids::IsBoundaryID... BoundaryIDTs>
bool serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > >::registered = (BoundaryIsIdentifiedBy<BoundaryT, BoundaryIDTs> || ...) [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/resolver.hpp

10.46 serif::discretization::domain::BoundarySet< BoundaryTs > Struct Template Reference

#include <concepts.hpp>

Static Public Attributes

- static constexpr std::size_t count = CountVariadicArguments<BoundaryTs...>()

10.46.1 Member Data Documentation

10.46.1.1 count

template<IsBoundary... BoundaryTs>
std::size_t serif::discretization::domain::BoundarySet< BoundaryTs >::count = CountVariadicArguments<BoundaryTs...>()
[static], [constexpr]

The documentation for this struct was generated from the following file:

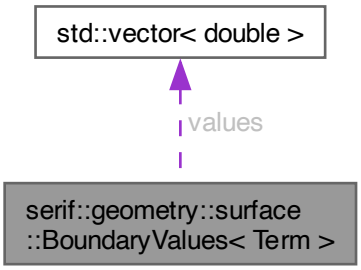
- src/include/serif/discretization/domain/concepts.hpp

10.47 serif::geometry::surface::BoundaryValues< Term > Struct Template Reference

Values of one scalar term at an ordered collection of boundary points.

#include <samples.hpp>

Collaboration diagram for serif::geometry::surface::BoundaryValues< Term >:



Public Member Functions

- bool operator== (const BoundaryValues &) const =default

  Compare the actual sampled values, including their count, without a lossy checksum.

Public Attributes

- std::vector< double > values

10.47.1 Detailed Description

template<SurfaceScalarTerm Term>
struct serif::geometry::surface::BoundaryValues< Term >

Values of one scalar term at an ordered collection of boundary points.

Note

Every term in a preparation must use the same point ordering. Sampling and finite-element trace choices belong to the backend.

10.47.2 Member Function Documentation

10.47.2.1 operator==()

template<SurfaceScalarTerm Term>
bool serif::geometry::surface::BoundaryValues< Term >::operator== (
const BoundaryValues< Term > &) const [nodiscard], [default]

Compare the actual sampled values, including their count, without a lossy checksum.

10.47.3 Member Data Documentation

10.47.3.1 values

template<SurfaceScalarTerm Term>
std::vector<double> serif::geometry::surface::BoundaryValues< Term >::values

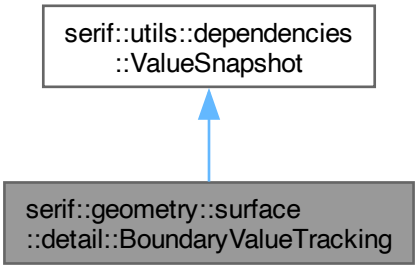
The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/samples.hpp

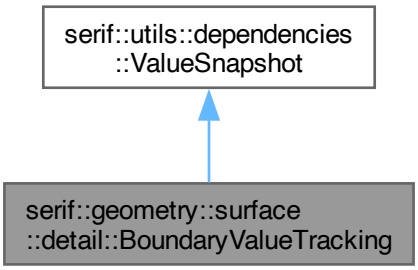
10.48 serif::geometry::surface::detail::BoundaryValueTracking Struct Reference

#include <samples.hpp>

Inheritance diagram for serif::geometry::surface::detail::BoundaryValueTracking:



Collaboration diagram for serif::geometry::surface::detail::BoundaryValueTracking:



Public Member Functions

- template<SurfaceScalarTerm Term>
  const void * identity (const BoundaryValues< Term > &) const

  Numerical samples have value semantics; rebuilding an equal context need not reevaluate the physics.

Public Member Functions inherited from serif::utils::dependencies::ValueSnapshot

- template<typename T>
  requires std::copy_constructible<T> && std::equality_comparable<T>
  T capture (const T &value) const

  Keep the accepted value independent of subsequent edits to a borrowed source.
- template<typename T>
  requires std::equality_comparable<T>
  bool matches (const T &captured, const T &value) const

  Compare content rather than a hash that could silently miss a change.

10.48.1 Member Function Documentation

10.48.1.1 identity()

template<SurfaceScalarTerm Term>
const void * serif::geometry::surface::detail::BoundaryValueTracking::identity (
const BoundaryValues< Term > &) const [inline], [nodiscard]

Numerical samples have value semantics; rebuilding an equal context need not reevaluate the physics.

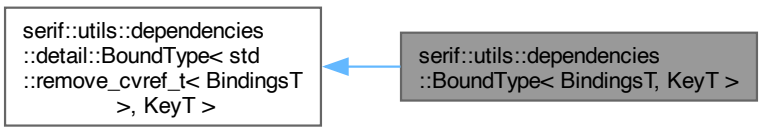
The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/samples.hpp

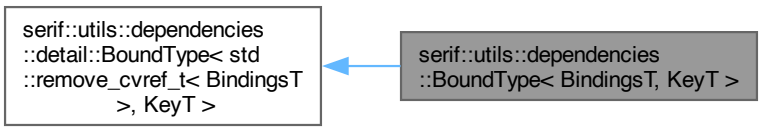
10.49 serif::utils::dependencies::BoundType< BindingsT, KeyT > Struct Template Reference

#include <binding.hpp>

Inheritance diagram for serif::utils::dependencies::BoundType< BindingsT, KeyT >:



Collaboration diagram for serif::utils::dependencies::BoundType< BindingsT, KeyT >:



The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/binding.hpp

10.50 serif::utils::dependencies::detail::BoundType< BindingsT, KeyT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/binding.hpp

10.51 serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT > Struct Template Reference

#include <binding.hpp>

Classes

- struct binding_tag
- struct collector

Public Types

- using target_type = decltype(select<KeyT>(collector{}))::type

Static Private Member Functions

- template<typename K, typename T>
  static binding_tag< K, T > select (binding_tag< K, T >)

10.51.1 Member Typedef Documentation

10.51.1.1 target_type

template<typename KeyT, IsBinding... BindingTs>
using serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::target_type = decltype(select<KeyT>(collector{}))↩
::type

10.51.2 Member Function Documentation

10.51.2.1 select()

template<typename KeyT, IsBinding... BindingTs>
template<typename K, typename T>
static binding_tag< K, T > serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::select (
binding_tag< K, T >) [static], [private]

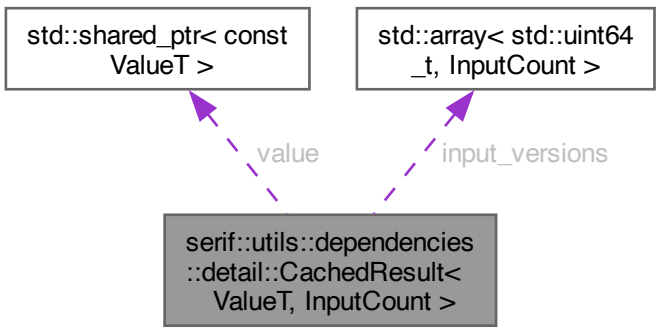
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/binding.hpp

10.52 serif::utils::dependencies::detail::CachedResult< ValueT, InputCount > Struct Template Reference

#include <cache_storage.hpp>

Collaboration diagram for serif::utils::dependencies::detail::CachedResult< ValueT, InputCount >:



Public Attributes

- std::shared_ptr< const ValueT > value
- std::array< std::uint64_t, InputCount > input_versions {}

10.52.1 Member Data Documentation

10.52.1.1 input_versions

template<typename ValueT, std::size_t InputCount>
std::array<std::uint64_t, InputCount> serif::utils::dependencies::detail::CachedResult< ValueT, InputCount >::input_versions {}

10.52.1.2 value

template<typename ValueT, std::size_t InputCount>
std::shared_ptr<const ValueT> serif::utils::dependencies::detail::CachedResult< ValueT, InputCount >::value

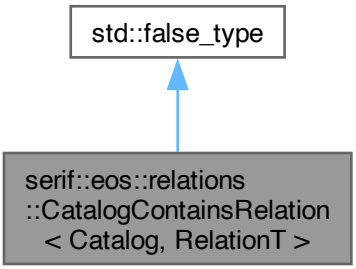
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_storage.hpp

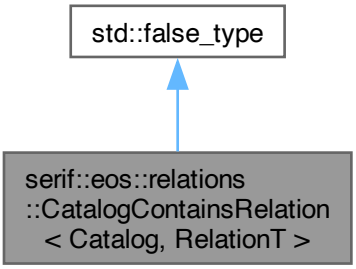
10.53 serif::eos::relations::CatalogContainsRelation< Catalog, RelationT > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::relations::CatalogContainsRelation< Catalog, RelationT >:



Collaboration diagram for serif::eos::relations::CatalogContainsRelation< Catalog, RelationT >:



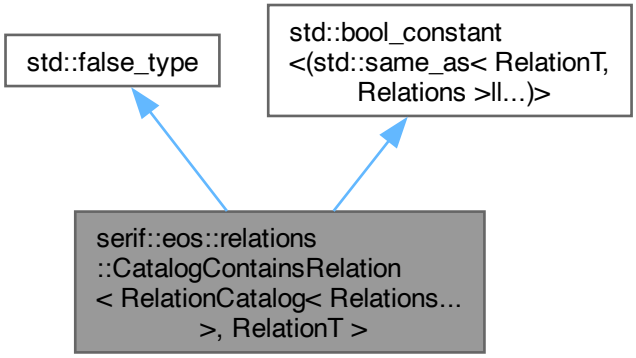
The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/traits.hpp

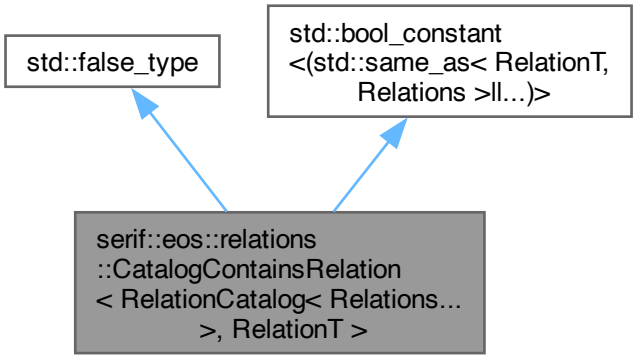
10.54 serif::eos::relations::CatalogContainsRelation< RelationCatalog< Relations... >, RelationT > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::relations::CatalogContainsRelation< RelationCatalog< Relations... >, RelationT >:



Collaboration diagram for serif::eos::relations::CatalogContainsRelation< RelationCatalog< Relations... >, RelationT >:



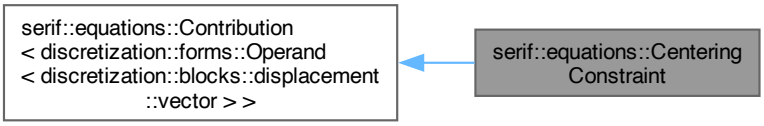
The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/traits.hpp

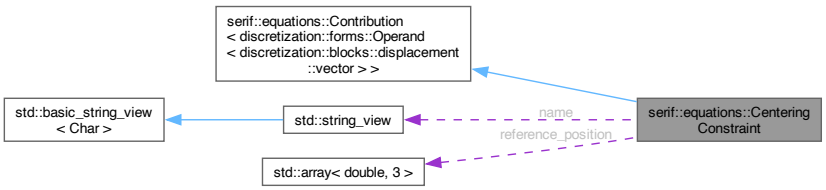
10.55 serif::equations::CenteringConstraint Struct Reference

#include <centering.hpp>

Inheritance diagram for serif::equations::CenteringConstraint:



Collaboration diagram for serif::equations::CenteringConstraint:



Public Types

- using Operands = utils::types::TypeList<TestOperand>

Public Types inherited from serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::displacement::vecto

- using Test
- using TestOperand
- using Support

Public Member Functions

- PointLinearization< Operands, discretization::forms::VectorValue > evaluate (const EquationPoint< Operands > &point) const

  Replace the three force rows by displacement and the identity derivative.

Static Public Attributes

- static constexpr std::string_view name = "centering constraint"
- static constexpr bool position_independent = true
- static constexpr std::array< double, 3 > reference_position {}

10.55.1 Detailed Description

Fix the material point at the computational origin: d(0) = 0. This is a translational gauge, not a center-of-mass constraint.

10.55.2 Member Typedef Documentation

10.55.2.1 Operands

using serif::equations::CenteringConstraint::Operands = utils::types::TypeList<TestOperand>

10.55.3 Member Function Documentation

10.55.3.1 evaluate()

PointLinearization< Operands, discretization::forms::VectorValue > serif::equations::CenteringConstraint::evaluate (const EquationPoint< Operands > & point) const [inline], [nodiscard]

Replace the three force rows by displacement and the identity derivative.

10.55.4 Member Data Documentation

10.55.4.1 name

std::string_view serif::equations::CenteringConstraint::name = "centering constraint" [static], [constexpr]

10.55.4.2 position_independent

bool serif::equations::CenteringConstraint::position_independent = true [static], [constexpr]

10.55.4.3 reference_position

std::array<double, 3> serif::equations::CenteringConstraint::reference_position {} [static], [constexpr]

The documentation for this struct was generated from the following file:

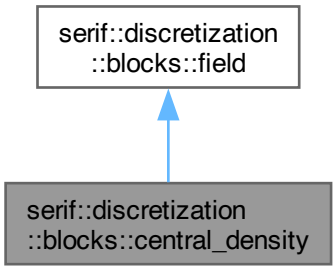
- src/include/serif/equations/centering.hpp

10.56 serif::discretization::blocks::central_density Struct Reference

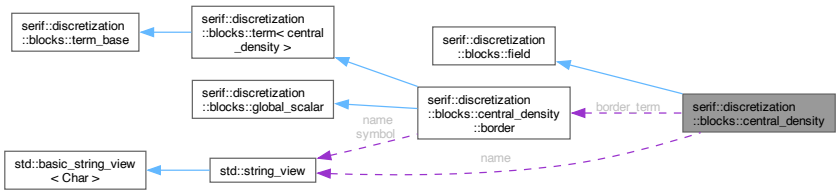
field used to inject a border row used to enforce a central density constraint

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::central_density:



Collaboration diagram for serif::discretization::blocks::central_density:



Classes

- struct border

  The border term represents the border row used to enforce a central density constraint. This is a global scalar as it is a single value over the entire domain.

Public Types

- using Terms = utils::types::TypeList<border>

  Type list of terms for compile time reflection and iteration over the terms of the field.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<central_density>()

  Compile time reflected name of the field.
- static constexpr border border_term {}

  Static constexpr instance of the border term. This is used to access the term in the discretization.

10.56.1 Detailed Description

field used to inject a border row used to enforce a central density constraint

10.56.2 Member Typedef Documentation

10.56.2.1 Terms

using serif::discretization::blocks::central_density::Terms = utils::types::TypeList<border>

Type list of terms for compile time reflection and iteration over the terms of the field.

10.56.3 Member Data Documentation

10.56.3.1 border_term

border serif::discretization::blocks::central_density::border_term {} [static], [constexpr]

Static constexpr instance of the border term. This is used to access the term in the discretization.

10.56.3.2 name

std::string_view serif::discretization::blocks::central_density::name = meta::type_name<central_density>() [static], [constexpr]

Compile time reflected name of the field.

The documentation for this struct was generated from the following file:

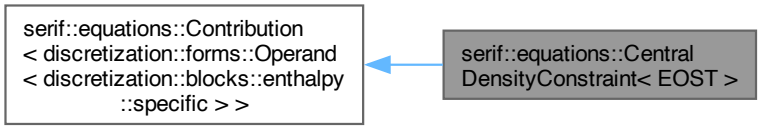
- src/include/serif/discretization/blocks/fields.hpp

10.57 serif::equations::CentralDensityConstraint< EOST > Class Template Reference

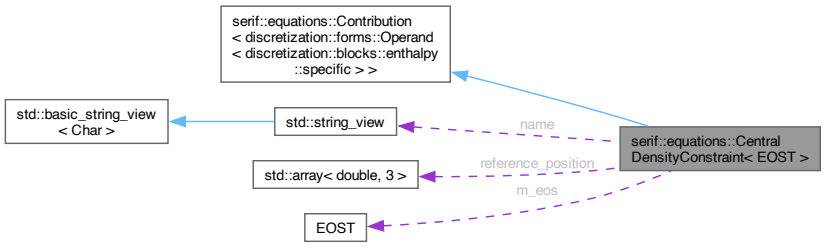
Prescribe central density through the continuous enthalpy: h(0) - h_EOS(rho_c) = 0.

#include <central_density.hpp>

Inheritance diagram for serif::equations::CentralDensityConstraint< EOST >:



Collaboration diagram for serif::equations::CentralDensityConstraint< EOST >:



Public Types

- using Operands = utils::types::TypeList<TestOperand>
- using Border = discretization::blocks::central_density::border

Public Types inherited from serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::enthalpy::specific >

- using Test
- using TestOperand
- using Support

Public Member Functions

- CentralDensityConstraint (EOST equation_of_state, dimensions::DensityValue target_density)

  Own the EOS and resolve the strictly positive prescribed density once.
- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const

  The border backend places this scalar condition in its global phase row.
- dimensions::DensityValue target_density () const noexcept

  Return the prescribed physical density.
- dimensions::SpecificEnthalpyValue target_enthalpy () const noexcept

  Return the equivalent EOS enthalpy target.
- dimensions::DensityValue density_from_enthalpy (dimensions::SpecificEnthalpyValue enthalpy) const

  Convert an achieved central enthalpy for physical diagnostics, when the inverse relation exists.

Static Public Attributes

- static constexpr std::string_view name = "central density constraint"
- static constexpr bool position_independent = true
- static constexpr bool affine = true
- static constexpr std::array< double, 3 > reference_position {}

Private Attributes

- EOST m_eos
- dimensions::DensityValue m_target_density
- double m_target_enthalpy

10.57.1 Detailed Description

template<eos::models::EOSModel EOST>
requires eos::models::EOSSupportingRelation<EOST, eos::relations::SpecificEnthalpyFromDensity>
class serif::equations::CentralDensityConstraint< EOST >

Prescribe central density through the continuous enthalpy: h(0) - h_EOS(rho_c) = 0.

10.57.2 Member Typedef Documentation

10.57.2.1 Border

template<eos::models::EOSModel EOST>
using serif::equations::CentralDensityConstraint< EOST >::Border = discretization::blocks::central_density::border

10.57.2.2 Operands

template<eos::models::EOSModel EOST>
using serif::equations::CentralDensityConstraint< EOST >::Operands = utils::types::TypeList<TestOperand>

10.57.3 Constructor & Destructor Documentation

10.57.3.1 CentralDensityConstraint()

template<eos::models::EOSModel EOST>
serif::equations::CentralDensityConstraint< EOST >::CentralDensityConstraint (
EOST equation_of_state,
dimensions::DensityValue target_density) [inline]

Own the EOS and resolve the strictly positive prescribed density once.

10.57.4 Member Function Documentation

10.57.4.1 density_from_enthalpy()

template<eos::models::EOSModel EOST>
dimensions::DensityValue serif::equations::CentralDensityConstraint< EOST >::density_from_enthalpy (
dimensions::SpecificEnthalpyValue enthalpy) const [inline], [nodiscard]

Convert an achieved central enthalpy for physical diagnostics, when the inverse relation exists.

10.57.4.2 evaluate()

template<eos::models::EOSModel EOST>
PointLinearization< Operands > serif::equations::CentralDensityConstraint< EOST >::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

The border backend places this scalar condition in its global phase row.

10.57.4.3 target_density()

template<eos::models::EOSModel EOST>
dimensions::DensityValue serif::equations::CentralDensityConstraint< EOST >::target_density () const [inline], [nodiscard], [noexcept]

Return the prescribed physical density.

10.57.4.4 target_enthalpy()

template<eos::models::EOSModel EOST>
dimensions::SpecificEnthalpyValue serif::equations::CentralDensityConstraint< EOST >::target_enthalpy () const [inline], [nodiscard], [noexcept]

Return the equivalent EOS enthalpy target.

10.57.5 Member Data Documentation

10.57.5.1 affine

template<eos::models::EOSModel EOST>
bool serif::equations::CentralDensityConstraint< EOST >::affine = true [static], [constexpr]

10.57.5.2 m_eos

template<eos::models::EOSModel EOST>
EOST serif::equations::CentralDensityConstraint< EOST >::m_eos [private]

10.57.5.3 m_target_density

template<eos::models::EOSModel EOST>
dimensions::DensityValue serif::equations::CentralDensityConstraint< EOST >::m_target_density [private]

10.57.5.4 m_target_enthalpy

template<eos::models::EOSModel EOST>
double serif::equations::CentralDensityConstraint< EOST >::m_target_enthalpy [private]

10.57.5.5 name

template<eos::models::EOSModel EOST>
std::string_view serif::equations::CentralDensityConstraint< EOST >::name = "central density constraint" [static], [constexpr]

10.57.5.6 position_independent

template<eos::models::EOSModel EOST>
bool serif::equations::CentralDensityConstraint< EOST >::position_independent = true [static], [constexpr]

10.57.5.7 reference_position

template<eos::models::EOSModel EOST>
std::array<double, 3> serif::equations::CentralDensityConstraint< EOST >::reference_position {} [static], [constexpr]

The documentation for this class was generated from the following file:

- src/include/serif/equations/central_density.hpp

10.58 serif::geometry::deformation::backend::mfem::volume_displacement_keys::CheckedBinding Struct Reference

#include <volume_displacement.hpp>

The documentation for this struct was generated from the following file:

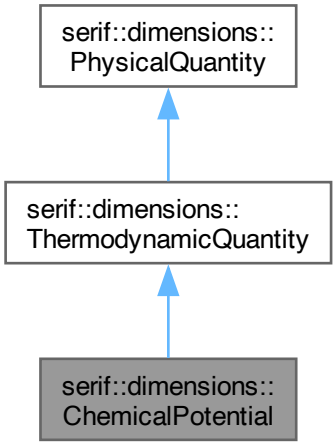
- src/include/serif/geometry/deformation/backend/mfem/volume_displacement.hpp

10.59 serif::dimensions::ChemicalPotential Struct Reference

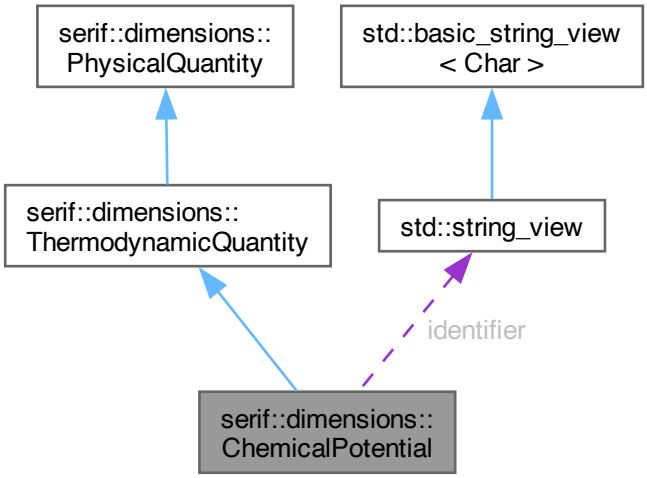
Dimension of Chemical Potential.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::ChemicalPotential:



Collaboration diagram for serif::dimensions::ChemicalPotential:



Static Public Attributes

- static constexpr std::string_view identifier = "chemical_potential"

10.59.1 Detailed Description

Dimension of Chemical Potential.

10.59.2 Member Data Documentation

10.59.2.1 identifier

std::string_view serif::dimensions::ChemicalPotential::identifier = "chemical_potential" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.60 serif::geometry::deformation::backend::mfem::volume_displacement_keys::Coefficients Struct Reference

#include <volume_displacement.hpp>

The documentation for this struct was generated from the following file:

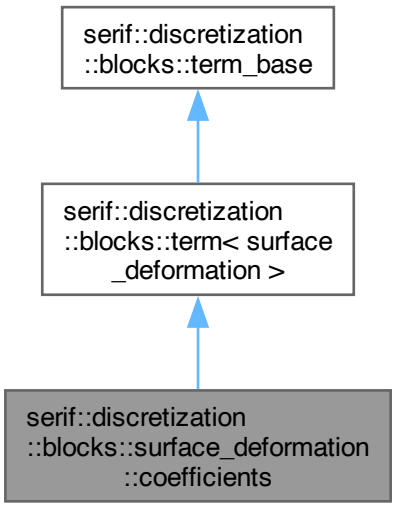
- src/include/serif/geometry/deformation/backend/mfem/volume_displacement.hpp

10.61 serif::discretization::blocks::surface_deformation::coefficients Struct Reference

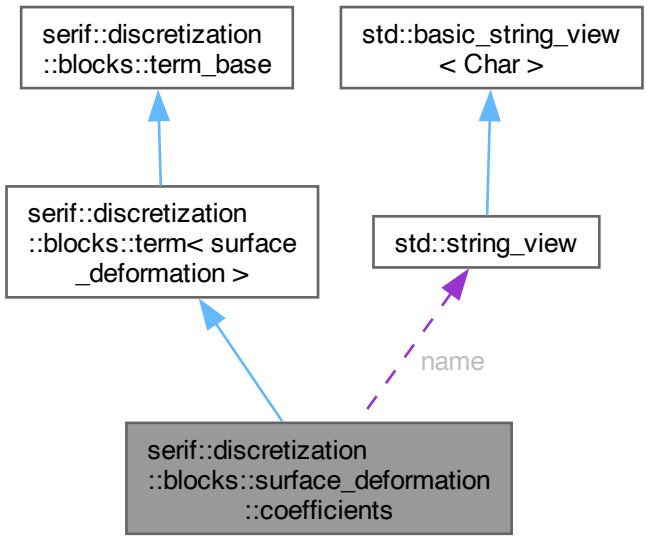
The coefficients term represents some set of coefficients associated with whatever surface representation is being used to represent the surface deformation. Note how we do not have a functional space, support, or a quantity. This is also where we associate the values with.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::surface_deformation::coefficients:



Collaboration diagram for serif::discretization::blocks::surface_deformation::coefficients:



Classes

- struct value

  The actual values of the coefficients for whatever surface representation is being used.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<coefficients>()

  Compile time reflected name of the term.

Additional Inherited Members

Public Types inherited from serif::discretization::blocks::term< surface_deformation >

- using field_type

10.61.1 Detailed Description

The coefficients term represents some set of coefficients associated with whatever surface representation is being used to represent the surface deformation. Note how we do not have a functional space, support, or a quantity. This is also where we associate the values with.

10.61.2 Member Data Documentation

10.61.2.1 name

std::string_view serif::discretization::blocks::surface_deformation::coefficients::name = meta::type_name<coefficients>() [static], [constexpr]

Compile time reflected name of the term.

The documentation for this struct was generated from the following file:

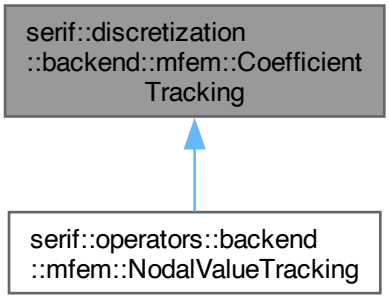
- src/include/serif/discretization/blocks/fields.hpp

10.62 serif::discretization::backend::mfem::CoefficientTracking Struct Reference

Track coefficient values using an owned snapshot and the identity of their binding.

#include <coefficient_tracking.hpp>

Inheritance diagram for serif::discretization::backend::mfem::CoefficientTracking:



Public Member Functions

- template<typename ViewT>
  bool matches (const ::mfem::Vector &captured, const ViewT &view) const
  Check whether the current coefficients equal a captured snapshot.
- template<typename ViewT>
  ::mfem::Vector capture (const ViewT &view)

Static Public Member Functions

- template<typename ViewT>
  static ::mfem::Vector capture (const ViewT &view)

  Copy the current coefficient values into an independent snapshot.
- template<typename ViewT>
  static const void * identity (const ViewT &view)

  Identify the persistent binding associated with the coefficients.

10.62.1 Detailed Description

Track coefficient values using an owned snapshot and the identity of their binding.

A coefficient vector describes numerical values, while its binding determines how those values correspond to discretization degrees of freedom. Both pieces of information matter when deciding whether cached work can be reused.

This policy provides three operations to the dependency system:

- capture copies the current coefficient values into an owned snapshot.
- matches compares current values against a previously captured snapshot.
- identity identifies the persistent binding associated with the values.

The vector's allocation is not part of its identity. A different vector containing equal coefficients can represent the same input when it uses the same binding. Conversely, equal coefficients associated with a different binding must not automatically reuse the previous calculation.

This policy supplies comparisons; the dependency system decides which computations must run.

Note

Value comparison uses floating-point equality without a tolerance. It does not check whether the coefficients are finite or admissible; those checks belong to input validation.

10.62.2 Member Function Documentation

10.62.2.1 capture() [1/2]

template<typename ViewT>
static ::mfem::Vector serif::discretization::backend::mfem::CoefficientTracking::capture (
const ViewT & view) [static], [nodiscard]

Copy the current coefficient values into an independent snapshot.

Subsequent changes to the source vector do not modify the returned snapshot. The entire supplied vector is copied; if the view contains only selected nodal coefficients, only those selected values are captured.

Template Parameters

ViewT	View type exposing its coefficients through a values member.

Parameters

in	view	View of the current coefficients.

Returns

An owned copy of the coefficient vector.

10.62.2.2 capture() [2/2]

```
template<typename ViewT>
::mfem::Vector serif::discretization::backend::mfem::CoefficientTracking::capture (
            const ViewT & view)
```

10.62.2.3 identity()

```
template<typename ViewT>
const void * serif::discretization::backend::mfem::CoefficientTracking::identity (
            const ViewT & view)   [static], [nodiscard]
```

Identify the persistent binding associated with the coefficients.

The returned address identifies the binding object, rather than the temporary view or the coefficient vector's storage. Creating another view of the same binding therefore preserves this identity.

Template Parameters

ViewT	View type exposing its persistent binding through a binding member.

Parameters

in	view	View referring to the coefficient binding.

Returns

The address of the binding object.

Precondition

The binding member refers to an object whose lifetime and address remain valid while the dependency system relies on this identity.

10.62.2.4 matches()

```
template<typename ViewT>
bool serif::discretization::backend::mfem::CoefficientTracking::matches (
                const ::mfem::Vector & captured,
                const ViewT & view) const    [nodiscard]
```

Check whether the current coefficients equal a captured snapshot.

The vectors must have the same size and equal corresponding entries. Two empty vectors match. Their storage addresses do not affect the comparison.

This method compares values only. Binding identity is supplied separately by identity and checked by the dependency system.

Template Parameters

ViewT	View type exposing its coefficients through a values member.

Parameters

in	captured	Previously captured coefficient values.
in	view	View of the current coefficients.

Returns

True when the sizes and all corresponding values are equal.

The documentation for this struct was generated from the following file:

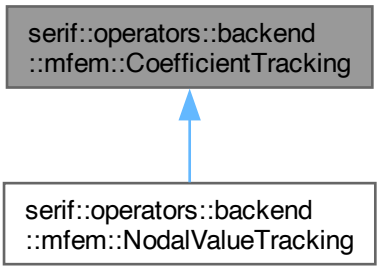
- src/include/serif/discretization/backend/mfem/coefficient_tracking.hpp

10.63 serif::operators::backend::mfem::CoefficientTracking Struct Reference

Track coefficient values using an owned snapshot and the identity of their binding.

#include <coefficient_tracking.hpp>

Inheritance diagram for serif::operators::backend::mfem::CoefficientTracking:



Public Member Functions

- template<typename ViewT>
  ::mfem::Vector capture (const ViewT &view)
- template<typename ViewT>
  bool matches (const ::mfem::Vector &captured, const ViewT &view) const

  Check whether the current coefficients equal a captured snapshot.

Static Public Member Functions

- template<typename ViewT>
  static ::mfem::Vector capture (const ViewT &view)

  Copy the current coefficient values into an independent snapshot.
- template<typename ViewT>
  static const void * identity (const ViewT &view)

  Identify the persistent binding associated with the coefficients.

10.63.1 Detailed Description

Track coefficient values using an owned snapshot and the identity of their binding.

A coefficient vector describes numerical values, while its binding determines how those values correspond to discretization degrees of freedom. Both pieces of information matter when deciding whether cached work can be reused.

This policy provides three operations to the dependency system:

- capture copies the current coefficient values into an owned snapshot.
- matches compares current values against a previously captured snapshot.
- identity identifies the persistent binding associated with the values.

The vector's allocation is not part of its identity. A different vector containing equal coefficients can represent the same input when it uses the same binding. Conversely, equal coefficients associated with a different binding must not automatically reuse the previous calculation.

This policy supplies comparisons; the dependency system decides which computations must run.

Note

Value comparison uses floating-point equality without a tolerance. It does not check whether the coefficients are finite or admissible; those checks belong to input validation.

10.63.2 Member Function Documentation

10.63.2.1 capture() [1/2]

template<typename ViewT>
static ::mfem::Vector serif::discretization::backend::mfem::CoefficientTracking::capture (
const ViewT & view) [static], [nodiscard]

Copy the current coefficient values into an independent snapshot.

Subsequent changes to the source vector do not modify the returned snapshot. The entire supplied vector is copied; if the view contains only selected nodal coefficients, only those selected values are captured.

Template Parameters

ViewT	View type exposing its coefficients through a values member.

Parameters

in	view	View of the current coefficients.

Returns

An owned copy of the coefficient vector.

10.63.2.2 capture() [2/2]

```
template<typename ViewT>
::mfem::Vector serif::discretization::backend::mfem::CoefficientTracking::capture (
            const ViewT & view)
```

10.63.2.3 identity()

```
template<typename ViewT>
const void * serif::discretization::backend::mfem::CoefficientTracking::identity (
            const ViewT & view)   [static], [nodiscard]
```

Identify the persistent binding associated with the coefficients.

The returned address identifies the binding object, rather than the temporary view or the coefficient vector's storage. Creating another view of the same binding therefore preserves this identity.

Template Parameters

ViewT	View type exposing its persistent binding through a binding member.

Parameters

in	view	View referring to the coefficient binding.

Returns

The address of the binding object.

Precondition

The binding member refers to an object whose lifetime and address remain valid while the dependency system relies on this identity.

10.63.2.4 matches()

```
template<typename ViewT>
bool serif::discretization::backend::mfem::CoefficientTracking::matches (
                const ::mfem::Vector & captured,
                const ViewT & view) const   [nodiscard]
```

Check whether the current coefficients equal a captured snapshot.

The vectors must have the same size and equal corresponding entries. Two empty vectors match. Their storage addresses do not affect the comparison.

This method compares values only. Binding identity is supplied separately by identity and checked by the dependency system.

Template Parameters

ViewT	View type exposing its coefficients through a values member.

Parameters

in	captured	Previously captured coefficient values.
in	view	View of the current coefficients.

Returns

True when the sizes and all corresponding values are equal.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/backend/mfem/coefficient_tracking.hpp

10.64 serif::operators::backend::mfem::CoefficientView< BindingT > Struct Template Reference

#include <term_coefficients.hpp>

Public Types

- using Term = BindingT::Term

Public Attributes

- const BindingT & binding
- const ::mfem::Vector & values

10.64.1 Member Typedef Documentation

10.64.1.1 Term

template<typename BindingT>
using serif::operators::backend::mfem::CoefficientView< BindingT >::Term = BindingT::Term

10.64.2 Member Data Documentation

10.64.2.1 binding

template<typename BindingT>
const BindingT& serif::operators::backend::mfem::CoefficientView< BindingT >::binding

10.64.2.2 values

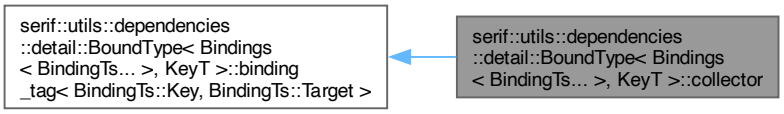
template<typename BindingT>
const ::mfem::Vector& serif::operators::backend::mfem::CoefficientView< BindingT >::values

The documentation for this struct was generated from the following file:

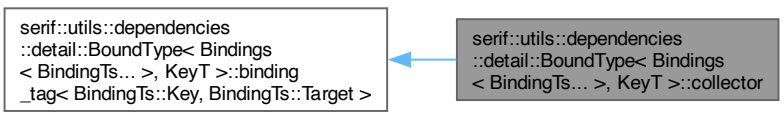
- src/include/serif/operators/backend/mfem/term_coefficients.hpp

10.65 serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::collector Struct Reference

Inheritance diagram for serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::collector:



Collaboration diagram for serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::collector:



Additional Inherited Members

Public Types inherited from
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< Binding

- using type

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/binding.hpp

10.66 serif::operators::detail::Columns< ListT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

10.67 serif::operators::detail::Columns< utils::types::TypeList< O... > > Struct Template Reference

#include <plan.hpp>

Public Types

- using type = typename utils::dependencies::CombinedKeyList<utils::dependencies::KeyList<discretization::forms::Oper
O::term_type>>...>::Keys

10.67.1 Member Typedef Documentation

10.67.1.1 type

template<typename... O>
using serif::operators::detail::Columns< utils::types::TypeList< O... > >::type = typename utils::dependencies::CombinedKeyList<utils::dependenci
O::term_type>>...>::Keys

The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

10.68 serif::operators::backend::mfem::CompactificationCoefficients Struct Reference

#include <volume_inputs.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_inputs.hpp

10.69 serif::mapping::CompactificationPointData Struct Reference

#include <data.hpp>

Public Attributes

- double compactification_coordinate {0.0}
- mfem::Vector coordinate_gradient

10.69.1 Member Data Documentation

10.69.1.1 compactification_coordinate

double serif::mapping::CompactificationPointData::compactification_coordinate {0.0}

10.69.1.2 coordinate_gradient

mfem::Vector serif::mapping::CompactificationPointData::coordinate_gradient

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/data.hpp

10.70 serif::utils::CompensatedSum Class Reference

Neumaier summation retains small terms when a scalar reduction contains cancellation.

#include <compensated_sum.hpp>

Public Member Functions

- void add (const double value)
- double value () const

Private Attributes

- double m_sum {0.0}
- double m_correction {0.0}

10.70.1 Detailed Description

Neumaier summation retains small terms when a scalar reduction contains cancellation.

10.70.2 Member Function Documentation

10.70.2.1 add()

void serif::utils::CompensatedSum::add (
const double value) [inline]

10.70.2.2 value()

double serif::utils::CompensatedSum::value () const [inline], [nodiscard]

10.70.3 Member Data Documentation

10.70.3.1 m_correction

double serif::utils::CompensatedSum::m_correction {0.0} [private]

10.70.3.2 m_sum

double serif::utils::CompensatedSum::m_sum {0.0} [private]

The documentation for this class was generated from the following file:

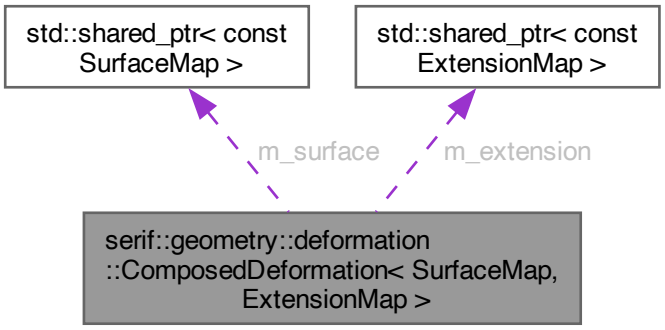
- src/include/serif/utils/misc/compensated_sum.hpp

10.71 serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap > Class Template Reference

Compose shape-to-surface and surface-to-volume maps with the complete derivative chain rule.

#include <composition.hpp>

Collaboration diagram for serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >:



Classes

- struct Workspace

Public Types

- using InputSpace = typename SurfaceMap::InputSpace
- using OutputSpace = typename ExtensionMap::OutputSpace

Public Member Functions

- std::size_t input_size () const noexcept

  Keep solver coordinates in the surface model's native parameter space.
- std::size_t output_size () const noexcept

  Expose the displacement field size consumed by the existing geometry mapper.
- Workspace make_workspace () const

  Reserve every intermediate once, outside repeated Newton and Krylov actions.
- DeformationStatus build_displacement (std::span< const double > parameters, std::span< double > displacement, Workspace &workspace) const

  Move the surface first, then extend that same vector trace into the volume.

- DeformationStatus apply_jacobian (std::span< const double > parameters, std::span< const double > direction, std::span< double > displacement_direction, Workspace &workspace) const

  Preserve nonlinear chain rules while avoiding an extra primal evaluation for fixed linear maps.
- DeformationStatus apply_jacobian_transpose (std::span< const double > parameters, std::span< const double > displacement_dual, std::span< double > parameter_dual, Workspace &workspace) const

  Reverse both factors in the forward composition, preserving their work pairing.
- DeformationStatus apply_pullback_derivative (std::span< const double > parameters, std::span< const double > direction, std::span< const double > displacement_dual, std::span< double > parameter_dual, Workspace &workspace) const

  Retain both the extension curvature and the derivative of the surface transpose.
- const SurfaceMap & surface_map () const noexcept

  Expose immutable surface mathematics for reference metrics and diagnostics.
- const ExtensionMap & extension_map () const noexcept

  Expose the immutable volume extension without sharing action workspaces.

Static Public Attributes

- static constexpr bool is_linear = is_linear_deformation<SurfaceMap> && is_linear_deformation<ExtensionMap>

Private Member Functions

- ComposedDeformation (std::shared_ptr< const SurfaceMap > surface, std::shared_ptr< const ExtensionMap > extension)

  Only the checked composition factory may establish factor compatibility.

Private Attributes

- std::shared_ptr< const SurfaceMap > m_surface
- std::shared_ptr< const ExtensionMap > m_extension

Friends

- template<DeformationMap First, DeformationMap Second>
  requires std::same_as<typename First::OutputSpace, typename Second::InputSpace>
  std::expected< ComposedDeformation< First, Second >, DeformationFailure > compose_deformation (std::shared_ptr< const First >, std::shared_ptr< const Second >)

10.71.1 Detailed Description

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
class serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >

Compose shape-to-surface and surface-to-volume maps with the complete derivative chain rule.

Note

Numerical factors are immutable and owned. Workspaces belong to callers and must not overlap public inputs or outputs; this class has no preparation state.

10.71.2 Member Typedef Documentation

10.71.2.1 InputSpace

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
using serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::InputSpace = typename SurfaceMap::InputSpace

10.71.2.2 OutputSpace

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
using serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::OutputSpace = typename ExtensionMap::OutputSpace

10.71.3 Constructor & Destructor Documentation

10.71.3.1 ComposedDeformation()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::ComposedDeformation (
std::shared_ptr< const SurfaceMap > surface,
std::shared_ptr< const ExtensionMap > extension) [inline], [private]

Only the checked composition factory may establish factor compatibility.

10.71.4 Member Function Documentation

10.71.4.1 apply_jacobian()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
DeformationStatus serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::apply_jacobian (
std::span< const double > parameters,
std::span< const double > direction,
std::span< double > displacement_direction,
Workspace & workspace) const [inline], [nodiscard]

Preserve nonlinear chain rules while avoiding an extra primal evaluation for fixed linear maps.

10.71.4.2 apply_jacobian_transpose()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
DeformationStatus serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::apply_jacobian_transpose (
std::span< const double > parameters,
std::span< const double > displacement_dual,
std::span< double > parameter_dual,
Workspace & workspace) const [inline], [nodiscard]

Reverse both factors in the forward composition, preserving their work pairing.

10.71.4.3 apply_pullback_derivative()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
DeformationStatus serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::apply_pullback_derivative (
std::span< const double > parameters,
std::span< const double > direction,
std::span< const double > displacement_dual,
std::span< double > parameter_dual,
Workspace & workspace) const [inline], [nodiscard]

Retain both the extension curvature and the derivative of the surface transpose.

10.71.4.4 build_displacement()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
DeformationStatus serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::build_displacement (
std::span< const double > parameters,
std::span< double > displacement,
Workspace & workspace) const [inline], [nodiscard]

Move the surface first, then extend that same vector trace into the volume.

10.71.4.5 extension_map()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
const ExtensionMap & serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::extension_map () const [inline], [nodiscard], [noexcept]

Expose the immutable volume extension without sharing action workspaces.

10.71.4.6 input_size()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
std::size_t serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::input_size () const [inline], [nodiscard], [noexcept]

Keep solver coordinates in the surface model's native parameter space.

10.71.4.7 make_workspace()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
Workspace serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::make_workspace () const [inline], [nodiscard]

Reserve every intermediate once, outside repeated Newton and Krylov actions.

10.71.4.8 output_size()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
std::size_t serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::output_size () const [inline], [nodiscard], [noexcept]

Expose the displacement field size consumed by the existing geometry mapper.

10.71.4.9 surface_map()

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
const SurfaceMap & serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::surface_map () const [inline], [nodiscard], [noexcept]

Expose immutable surface mathematics for reference metrics and diagnostics.

10.71.5 Friends And Related Symbol Documentation

10.71.5.1 compose_deformation

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
template<DeformationMap First, DeformationMap Second>
requires std::same_as<typename First::OutputSpace, typename Second::InputSpace>
std::expected< ComposedDeformation< First, Second >, DeformationFailure > compose_deformation (
std::shared_ptr< const First > ,
std::shared_ptr< const Second >) [friend]

10.71.6 Member Data Documentation

10.71.6.1 is_linear

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
bool serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::is_linear = is_linear_deformation<SurfaceMap> && is_linear_deformation<ExtensionMap> [static], [constexpr]

10.71.6.2 m_extension

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
std::shared_ptr<const ExtensionMap> serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::m_extension [private]

10.71.6.3 m_surface

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
std::shared_ptr<const SurfaceMap> serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::m_surface [private]

The documentation for this class was generated from the following file:

- src/include/serif/geometry/deformation/composition.hpp

10.72 serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT > Class Template Reference

#include <computation.hpp>

Public Types

- using Key = KeyT
- using Inputs = std::remove_cvref_t<InputsT>
- using Function = FunctionT

Public Member Functions

- Computation (FunctionT function)

  Own the callable; result-affecting mutable state belongs in the declared inputs.
- template<typename... Arguments>
  requires std::invocable<const FunctionT&, const Arguments&...>
  decltype(auto) evaluate (const Arguments &... arguments) const

  Supply inputs in declaration order and infer the result from the callable.

Private Attributes

- FunctionT m_function

10.72.1 Member Typedef Documentation

10.72.1.1 Function

template<typename KeyT, IsKeyList InputsT, typename FunctionT>
using serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >::Function = FunctionT

10.72.1.2 Inputs

template<typename KeyT, IsKeyList InputsT, typename FunctionT>
using serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >::Inputs = std::remove_cvref_t<InputsT>

10.72.1.3 Key

template<typename KeyT, IsKeyList InputsT, typename FunctionT>
using serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >::Key = KeyT

10.72.2 Constructor & Destructor Documentation

10.72.2.1 Computation()

template<typename KeyT, IsKeyList InputsT, typename FunctionT>
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >::Computation (
FunctionT function) [inline], [explicit]

Own the callable; result-affecting mutable state belongs in the declared inputs.

10.72.3 Member Function Documentation

10.72.3.1 evaluate()

template<typename KeyT, IsKeyList InputsT, typename FunctionT>
template<typename... Arguments>
requires std::invocable<const FunctionT&, const Arguments&...>
decltype(auto) serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >::evaluate (
const Arguments &... arguments) const [inline], [nodiscard]

Supply inputs in declaration order and infer the result from the callable.

10.72.4 Member Data Documentation

10.72.4.1 m_function

template<typename KeyT, IsKeyList InputsT, typename FunctionT>
FunctionT serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >::m_function [private]

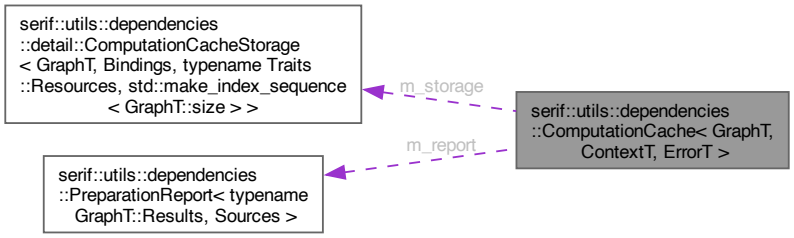
The documentation for this class was generated from the following file:

- src/include/serif/utils/dependencies/computation.hpp

10.73 serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT > Class Template Reference

#include <computation_cache.hpp>

Collaboration diagram for serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >:



Public Types

- using Traits = detail::ComputationCacheTraits<GraphT, ContextT, ErrorT>
- using Bindings = typename Traits::Bindings
- using Sources = typename Traits::Sources
- using Report = PreparationReport<typename GraphT::Results, Sources>
- using Failure = PreparationFailure<ErrorT>

Public Member Functions

- ComputationCache (GraphT graph)

  Own a graph and its accepted cache; contexts are supplied separately to each preparation.
- ComputationCache (const ComputationCache &)=delete
- ComputationCache & operator= (const ComputationCache &)=delete
- ComputationCache (ComputationCache &&other)

  Transfer accepted results while leaving the original object unprepared.
- ComputationCache & operator= (ComputationCache &&)=delete
- template<IsKeyList RequestedT = GraphT::Results>
  requires detail::key_list_subset_v<std::remove_cvref_t<RequestedT>, typename GraphT::Results>
  std::expected< Report, Failure > try_prepare (const ContextT &context, RequestedT={}, const PreparationMode mode=PreparationMode::reuse_unchanged)

  Prepare the requested results and their prerequisites, publishing only a complete successful candidate.
- template<typename KeyT>
  requires detail::contains_key_v<KeyT, typename GraphT::Results>
  bool contains () const

  Test whether a result belongs to the latest successful request, including its prerequisites.
- template<typename KeyT>
  requires detail::contains_key_v<KeyT, typename GraphT::Results>
  const auto & get () const

Read a frozen result. References remain valid until the next preparation or invalidation.

- const Report & report () const noexcept

  Report reuse and rebuilding from the latest successful preparation.

- bool ready () const noexcept

  A failed preparation makes all result access unavailable until a subsequent success.

- std::size_t preparation_count () const noexcept

  Count successful preparations, including requests whose results were all reused.

- void invalidate () noexcept

  Discard accepted snapshots and results, requiring fresh preparation on the next request.

Private Types

- using Storage = detail::ComputationCacheStorage<GraphT, Bindings, typename Traits::Resources, std::make_index_sequence<GraphT::size>>
- using ResultMask = typename GraphT::Dependencies::Mask
- using SourceMask = std::array<bool, source_count>

Private Member Functions

- template<typename KeyT>
  void inspect_source (const ContextT &context, Storage &candidate, Report &report, SourceMask &inspected, const PreparationMode mode) const

  Inspect each required source once; other context objects need no snapshot policy.

- template<typename KeyT>
  const auto & input (const ContextT &context, const Storage &candidate) const

  Computations read either a supplied source or a prerequisite already prepared in this candidate.

- template<std::size_t Index, typename... Keys>
  std::expected< void, Failure > prepare_one (const ContextT &context, Storage &candidate, Report &report, SourceMask &inspected, const PreparationMode mode, KeyList< Keys... >) const

  Reuse a result only when every input version matches its accepted version.

- template<std::size_t Position>
  std::expected< void, Failure > prepare_in_order (const ContextT &context, Storage &candidate, Report &report, SourceMask &inspected, const ResultMask &required, PreparationMode mode) const

  Follow the compile-time order while skipping branches outside the requested closure.

Static Private Member Functions

- template<typename KeyT>
  static consteval std::size_t version_index ()

  Share one version namespace between source snapshots and computed results.

Private Attributes

- GraphT m_graph
- Storage m_storage
- Report m_report
- ResultMask m_active {}
- bool m_ready {false}
- std::size_t m_preparation_count {0}

Static Private Attributes

- static constexpr std::size_t source_count = detail::KeyListSize<Sources>::value

10.73.1 Member Typedef Documentation

10.73.1.1 Bindings

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
using serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::Bindings = typename Traits::Bindings

10.73.1.2 Failure

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
using serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::Failure = PreparationFailure<ErrorT>

10.73.1.3 Report

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
using serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::Report = PreparationReport<typename GraphT::Results, Sources>

10.73.1.4 ResultMask

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
using serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::ResultMask = typename GraphT::↩ Dependencies::Mask [private]

10.73.1.5 SourceMask

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
using serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::SourceMask = std::array<bool, source_count> [private]

10.73.1.6 Sources

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
using serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::Sources = typename Traits::Sources

10.73.1.7 Storage

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
using serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::Storage = detail::ComputationCacheStorage<GraphT, Bindings, typename Traits::Resources, std::make_index_sequence<GraphT::size>> [private]

10.73.1.8 Traits

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
using serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::Traits = detail::ComputationCacheTraits<GraphT, ContextT, ErrorT>

10.73.2 Constructor & Destructor Documentation

10.73.2.1 ComputationCache() [1/3]

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::ComputationCache (
GraphT graph) [inline], [explicit]

Own a graph and its accepted cache; contexts are supplied separately to each preparation.

10.73.2.2 ComputationCache() [2/3]

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::ComputationCache (
const ComputationCache< GraphT, ContextT, ErrorT > &) [delete]

10.73.2.3 ComputationCache() [3/3]

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::ComputationCache (
ComputationCache< GraphT, ContextT, ErrorT > && other) [inline]

Transfer accepted results while leaving the original object unprepared.

10.73.3 Member Function Documentation

10.73.3.1 contains()

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
template<typename KeyT>
requires detail::contains_key_v<KeyT, typename GraphT::Results>
bool serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::contains () const [inline], [nodiscard]

Test whether a result belongs to the latest successful request, including its prerequisites.

10.73.3.2 get()

```
template<typename GraphT, typename ContextT, typename ErrorT = std::string>
template<typename KeyT>
requires detail::contains_key_v<KeyT, typename GraphT::Results>
const auto & serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::get () const    [inline], [nodiscard]
```

Read a frozen result. References remain valid until the next preparation or invalidation.

10.73.3.3 input()

```
template<typename GraphT, typename ContextT, typename ErrorT = std::string>
template<typename KeyT>
const auto & serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::input (
                const ContextT & context,
                const Storage & candidate) const    [inline], [nodiscard], [private]
```

Computations read either a supplied source or a prerequisite already prepared in this candidate.

10.73.3.4 inspect_source()

```
template<typename GraphT, typename ContextT, typename ErrorT = std::string>
template<typename KeyT>
void serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::inspect_source (
                const ContextT & context,
                Storage & candidate,
                Report & report,
                SourceMask & inspected,
                const PreparationMode mode) const    [inline], [private]
```

Inspect each required source once; other context objects need no snapshot policy.

10.73.3.5 invalidate()

```
template<typename GraphT, typename ContextT, typename ErrorT = std::string>
void serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::invalidate ()    [inline], [noexcept]
```

Discard accepted snapshots and results, requiring fresh preparation on the next request.

10.73.3.6 operator=() [1/2]

```
template<typename GraphT, typename ContextT, typename ErrorT = std::string>
ComputationCache & serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::operator= (
                ComputationCache< GraphT, ContextT, ErrorT > && )    [delete]
```

10.73.3.7 operator=() [2/2]

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
ComputationCache & serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::operator= (
const ComputationCache< GraphT, ContextT, ErrorT > &) [delete]

10.73.3.8 preparation_count()

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
std::size_t serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::preparation_count () const [inline], [nodiscard], [noexcept]

Count successful preparations, including requests whose results were all reused.

10.73.3.9 prepare_in_order()

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
template<std::size_t Position>
std::expected< void, Failure > serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::prepare_in_order (
const ContextT & context,
Storage & candidate,
Report & report,
SourceMask & inspected,
const ResultMask & required,
PreparationMode mode) const [inline], [nodiscard], [private]

Follow the compile-time order while skipping branches outside the requested closure.

10.73.3.10 prepare_one()

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
template<std::size_t Index, typename... Keys>
std::expected< void, Failure > serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::prepare_one (
const ContextT & context,
Storage & candidate,
Report & report,
SourceMask & inspected,
const PreparationMode mode,
KeyList< Keys... >) const [inline], [nodiscard], [private]

Reuse a result only when every input version matches its accepted version.

10.73.3.11 ready()

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
bool serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::ready () const [inline], [nodiscard], [noexcept]

A failed preparation makes all result access unavailable until a subsequent success.

10.73.3.12 report()

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
const Report & serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::report () const [inline], [nodiscard], [noexcept]

Report reuse and rebuilding from the latest successful preparation.

10.73.3.13 try_prepare()

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
template<IsKeyList RequestedT = GraphT::Results>
requires detail::key_list_subset_v<std::remove_cvref_t<RequestedT>, typename GraphT::Results>
std::expected< Report, Failure > serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::try_prepare (
const ContextT & context,
RequestedT = {},
const PreparationMode mode = PreparationMode::reuse_unchanged) [inline], [nodiscard]

Prepare the requested results and their prerequisites, publishing only a complete successful candidate.

10.73.3.14 version_index()

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
template<typename KeyT>
static consteval std::size_t serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::version_index () [inline], [static], [nodiscard], [consteval], [private]

Share one version namespace between source snapshots and computed results.

10.73.4 Member Data Documentation

10.73.4.1 m_active

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
ResultMask serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::m_active {} [private]

10.73.4.2 m_graph

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
GraphT serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::m_graph [private]

10.73.4.3 m_preparation_count

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
std::size_t serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::m_preparation_count {0} [private]

10.73.4.4 m_ready

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
bool serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::m_ready {false} [private]

10.73.4.5 m_report

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
Report serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::m_report [private]

10.73.4.6 m_storage

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
Storage serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::m_storage [private]

10.73.4.7 source_count

template<typename GraphT, typename ContextT, typename ErrorT = std::string>
std::size_t serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >::source_count = detail::KeyListSize<Sources>::value [static], [constexpr], [private]

The documentation for this class was generated from the following file:

- src/include/serif/utils/dependencies/computation_cache.hpp

10.74 serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, ResourcesT, IndicesT > Struct Template Reference

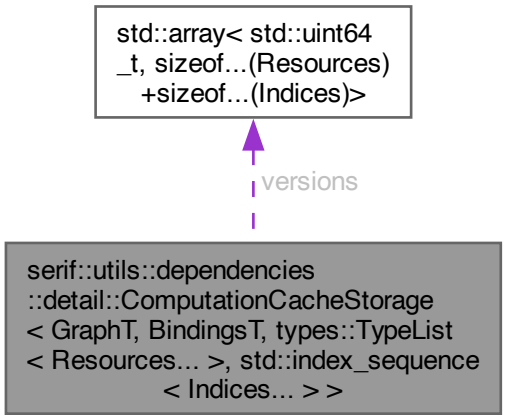
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_storage.hpp

10.75 serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > > Struct Template Reference

#include <cache_storage.hpp>

Collaboration diagram for serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > >:



Public Attributes

- std::tuple< SourceSnapshot< Resources >... > sources
- std::tuple< CachedResult< bound_type_t< BindingsT, typename GraphT::template Computation↩ At< Indices >::Key >, KeyListSize< typename GraphT::template ComputationAt< Indices >::Inputs >::value >... > results
- std::array< std::uint64_t, sizeof...(Resources)+sizeof...(Indices)> versions {}

10.75.1 Member Data Documentation

10.75.1.1 results

template<typename GraphT, typename BindingsT, typename... Resources, std::size_t... Indices>
std::tuple<CachedResult<bound_type_t<BindingsT, typename GraphT::template ComputationAt<Indices>::Key>, KeyListSize<typename GraphT::template ComputationAt<Indices>::Inputs>::value>...> serif::utils::dependencies::detail::ComputationCacheStora
GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > >::results

10.75.1.2 sources

template<typename GraphT, typename BindingsT, typename... Resources, std::size_t... Indices>
std::tuple<SourceSnapshot<Resources>...> serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > >::sources

10.75.1.3 versions

template<typename GraphT, typename BindingsT, typename... Resources, std::size_t... Indices>
std::array<std::uint64_t, sizeof...(Resources) + sizeof...(Indices)> serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > >::versions {}

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_storage.hpp

10.76 serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, Valid > Struct Template Reference

#include <cache_traits.hpp>

Static Public Attributes

- static constexpr bool valid = false

10.76.1 Member Data Documentation

10.76.1.1 valid

template<typename GraphT, typename ContextT, typename ErrorT, bool Valid = ComputationGraphFor<GraphT, Context↩T>>
bool serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, Valid >::valid = false [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_traits.hpp

10.77 serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true > Struct Template Reference

#include <cache_traits.hpp>

Public Types

- using Bindings = typename ComputationGraphSchema<GraphT, ContextT>::ResolvedBindings
- using Resources = UsedResources<GraphT, ContextT>
- using Sources = typename ResourceKeys<Resources>::type

Static Public Attributes

- static constexpr bool valid
- static constexpr bool valid

10.77.1 Member Typedef Documentation

10.77.1.1 Bindings

template<typename GraphT, typename ContextT, typename ErrorT>
using serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >::Bindings = typename ComputationGraphSchema<GraphT, ContextT>::ResolvedBindings

10.77.1.2 Resources

template<typename GraphT, typename ContextT, typename ErrorT>
using serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >::Resources = UsedResources<GraphT, ContextT>

10.77.1.3 Sources

template<typename GraphT, typename ContextT, typename ErrorT>
using serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >::Sources = typename ResourceKeys<Resources>::type

10.77.2 Member Data Documentation

10.77.2.1 valid [1/2]

bool serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, Valid >::valid [static], [constexpr]

10.77.2.2 valid [2/2]

template<typename GraphT, typename ContextT, typename ErrorT>
bool serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >::valid [static], [constexpr]

Initial value:

```
= ResourceKeys<Resources>::trackable && std::is_object_v<ErrorT> && std::move_constructible<ErrorT> &&
        compatible_errors<GraphT, Bindings, ErrorT>(std::make_index_sequence<GraphT::size>{})
```

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_traits.hpp

10.78 serif::utils::dependencies::ComputationGraph< Computations > Class Template Reference

#include <computation_graph.hpp>

Public Types

- using Results = KeyList<typename Computations::Key...>
- using Inputs = CombinedKeyList<typename Computations::Inputs...>
- using Dependencies = detail::Graph<Computations...>
- template<std::size_t Index>
  using ComputationAt = std::tuple_element_t<Index, std::tuple<Computations...>>

Public Member Functions

- ComputationGraph (Computations... computations)

  Accept computations in any order; the graph determines their evaluation order.
- template<std::size_t Index>
  requires (Index < size)
  const auto & computation () const

  Read a callable without permitting its stored configuration to change between preparations.

Static Public Member Functions

- template<typename KeyT>
  requires detail::contains_key_v<KeyT, Results>
  static consteval std::size_t index ()

  Convert a known result key to its stable position in the graph.

Static Public Attributes

- static constexpr std::size_t size = sizeof...(Computations)

Private Attributes

- std::tuple< Computations... > m_computations

10.78.1 Member Typedef Documentation

10.78.1.1 ComputationAt

template<IsComputation... Computations>
template<std::size_t Index>
using serif::utils::dependencies::ComputationGraph< Computations >::ComputationAt = std::tuple_element_t<Index, std::tuple<Computations...>>

10.78.1.2 Dependencies

template<IsComputation... Computations>
using serif::utils::dependencies::ComputationGraph< Computations >::Dependencies = detail::Graph<Computations...>

10.78.1.3 Inputs

template<IsComputation... Computations>
using serif::utils::dependencies::ComputationGraph< Computations >::Inputs = CombinedKeyList<typename Computations::Inputs...>

10.78.1.4 Results

template<IsComputation... Computations>
using serif::utils::dependencies::ComputationGraph< Computations >::Results = KeyList<typename Computations::Key...>

10.78.2 Constructor & Destructor Documentation

10.78.2.1 ComputationGraph()

template<IsComputation... Computations>
serif::utils::dependencies::ComputationGraph< Computations >::ComputationGraph (
Computations... computations) [inline], [explicit]

Accept computations in any order; the graph determines their evaluation order.

10.78.3 Member Function Documentation

10.78.3.1 computation()

template<IsComputation... Computations>
template<std::size_t Index>
requires (Index < size)
const auto & serif::utils::dependencies::ComputationGraph< Computations >::computation () const [inline], [nodiscard]

Read a callable without permitting its stored configuration to change between preparations.

10.78.3.2 index()

template<IsComputation... Computations>
template<typename KeyT>
requires detail::contains_key_v<KeyT, Results>
static consteval std::size_t serif::utils::dependencies::ComputationGraph< Computations >::index () [inline], [static], [nodiscard], [consteval]

Convert a known result key to its stable position in the graph.

10.78.4 Member Data Documentation

10.78.4.1 m_computations

template<IsComputation... Computations>
std::tuple<Computations...> serif::utils::dependencies::ComputationGraph< Computations >::m_computations [private]

10.78.4.2 size

template<IsComputation... Computations>
std::size_t serif::utils::dependencies::ComputationGraph< Computations >::size = sizeof...(Computations) [static], [constexpr]

The documentation for this class was generated from the following file:

- src/include/serif/utils/dependencies/computation_graph.hpp

10.79 serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, Independent > Struct Template Reference

#include <schema.hpp>

Static Public Attributes

- static constexpr bool valid = false

10.79.1 Member Data Documentation

10.79.1.1 valid

template<typename GraphT, typename ContextT, bool Independent = independent_results<typename ContextT::↩
Bindings>(typename GraphT::Results{})>
bool serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, Independent >::valid = false [static], [constexpr]

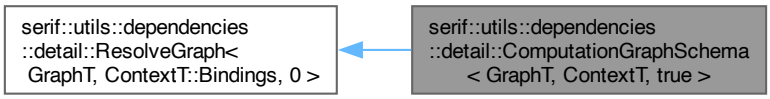
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

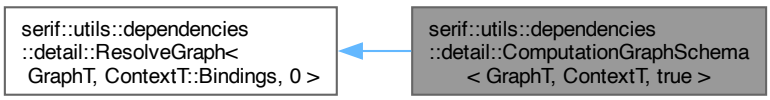
10.80 serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, true > Struct Template Reference

#include <schema.hpp>

Inheritance diagram for serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, true >:



Collaboration diagram for serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, true >:



Static Public Attributes

- static constexpr bool valid

10.80.1 Member Data Documentation

10.80.1.1 valid

bool serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, Independent >::valid [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

10.81 serif::utils::dependencies::ComputationStore< EntryTs > Class Template Reference

Own independent computation caches under compile-time role keys; this store does not connect their graphs.

#include <computation_store.hpp>

Public Types

- using Bindings = dependencies::Bindings<Binding<typename EntryTs::Key, typename EntryTs::↩ Target>...>

Public Member Functions

- ComputationStore (EntryTs... entries)

  Consumers may borrow entries while this store or a moved-to successor remains alive.
- ComputationStore (const ComputationStore &)=delete

  Duplicating ownership would give consumers ambiguous cache identities.
- ComputationStore & operator= (const ComputationStore &)=delete
- ComputationStore (ComputationStore &&) noexcept=default

  Moving a store preserves the addresses of all its computations.
- ComputationStore & operator= (ComputationStore &&)=delete

  Do not replace computations that existing consumers may still reference.
- template<typename KeyT>
  requires BindingsWithBinding<Bindings, KeyT>
  auto & get () &

  Retrieve a computation by its model role, independently of its implementation type.
- template<typename KeyT>
  requires BindingsWithBinding<Bindings, KeyT>
  const auto & get () const &

  Const owners expose read-only access to each computation.
- template<typename KeyT>
  auto get () &&=delete

  Borrowed computations must not outlive a temporary store.
- template<typename KeyT>
  auto get () const &&=delete

Private Attributes

- std::tuple< EntryTs... > m_entries

10.81.1 Detailed Description

template<IsOwnedComputation... EntryTs>
requires misc::traits::all_unique_types_v<typename EntryTs::Key...>
class serif::utils::dependencies::ComputationStore< EntryTs >

Own independent computation caches under compile-time role keys; this store does not connect their graphs.

10.81.2 Member Typedef Documentation

10.81.2.1 Bindings

template<IsOwnedComputation... EntryTs>
using serif::utils::dependencies::ComputationStore< EntryTs >::Bindings = dependencies::Bindings<Binding<typename EntryTs::Key, typename EntryTs::Target>...>

10.81.3 Constructor & Destructor Documentation

10.81.3.1 ComputationStore() [1/3]

template<IsOwnedComputation... EntryTs>
serif::utils::dependencies::ComputationStore< EntryTs >::ComputationStore (
EntryTs... entries) [inline], [explicit]

Consumers may borrow entries while this store or a moved-to successor remains alive.

10.81.3.2 ComputationStore() [2/3]

template<IsOwnedComputation... EntryTs>
serif::utils::dependencies::ComputationStore< EntryTs >::ComputationStore (
const ComputationStore< EntryTs > &) [delete]

Duplicating ownership would give consumers ambiguous cache identities.

10.81.3.3 ComputationStore() [3/3]

template<IsOwnedComputation... EntryTs>
serif::utils::dependencies::ComputationStore< EntryTs >::ComputationStore (
ComputationStore< EntryTs > &&) [default], [noexcept]

Moving a store preserves the addresses of all its computations.

10.81.4 Member Function Documentation

10.81.4.1 get() [1/4]

template<IsOwnedComputation... EntryTs>
template<typename KeyT>
requires BindingsWithBinding<Bindings, KeyT>
auto & serif::utils::dependencies::ComputationStore< EntryTs >::get () & [inline], [nodiscard]

Retrieve a computation by its model role, independently of its implementation type.

10.81.4.2 get() [2/4]

template<IsOwnedComputation... EntryTs>
template<typename KeyT>
auto serif::utils::dependencies::ComputationStore< EntryTs >::get () && [delete]

Borrowed computations must not outlive a temporary store.

10.81.4.3 get() [3/4]

template<IsOwnedComputation... EntryTs>
template<typename KeyT>
requires BindingsWithBinding<Bindings, KeyT>
const auto & serif::utils::dependencies::ComputationStore< EntryTs >::get () const & [inline], [nodiscard]

Const owners expose read-only access to each computation.

10.81.4.4 get() [4/4]

template<IsOwnedComputation... EntryTs>
template<typename KeyT>
auto serif::utils::dependencies::ComputationStore< EntryTs >::get () const && [delete]

10.81.4.5 operator=() [1/2]

template<IsOwnedComputation... EntryTs>
ComputationStore & serif::utils::dependencies::ComputationStore< EntryTs >::operator= (
ComputationStore< EntryTs > &&) [delete]

Do not replace computations that existing consumers may still reference.

10.81.4.6 operator=() [2/2]

template<IsOwnedComputation... EntryTs>
ComputationStore & serif::utils::dependencies::ComputationStore< EntryTs >::operator= (
const ComputationStore< EntryTs > &) [delete]

10.81.5 Member Data Documentation

10.81.5.1 m_entries

template<IsOwnedComputation... EntryTs>
std::tuple<EntryTs...> serif::utils::dependencies::ComputationStore< EntryTs >::m_entries [private]

The documentation for this class was generated from the following file:

- src/include/serif/utils/dependencies/computation_store.hpp

10.82 serif::utils::dependencies::ComputationUpdate< InputsT, ValueT > Struct Template Reference

#include <incremental_computation.hpp>

Collaboration diagram for serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >:



Public Member Functions

- template<typename KeyT>
  requires detail::contains_key_v<KeyT, InputsT>
  bool changed () const

  A new or forced calculation refreshes every input, even when values compare equal.

Public Attributes

- const ValueT * previous {nullptr}
- std::array< bool, detail::KeyListSize< InputsT >::value > changed_inputs {}
- bool rebuild {false}

10.82.1 Member Function Documentation

10.82.1.1 changed()

template<IsKeyList InputsT, typename ValueT>
template<typename KeyT>
requires detail::contains_key_v<KeyT, InputsT>
bool serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >::changed () const [inline], [nodiscard]

A new or forced calculation refreshes every input, even when values compare equal.

10.82.2 Member Data Documentation

10.82.2.1 changed_inputs

template<IsKeyList InputsT, typename ValueT>
std::array<bool, detail::KeyListSize<InputsT>::value> serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >↩
::changed_inputs {}

10.82.2.2 previous

template<IsKeyList InputsT, typename ValueT>
const ValueT* serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >::previous {nullptr}

10.82.2.3 rebuild

template<IsKeyList InputsT, typename ValueT>
bool serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >::rebuild {false}

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/incremental_computation.hpp

10.83 serif::utils::misc::concepts::Concat< Lists > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/concat.hpp

10.84 serif::utils::misc::concepts::Concat< types::TypeList< LhsTs... >, types::TypeList< RhsTs... >, Rest... > Struct Template Reference

#include <concat.hpp>

Public Types

- using Type = typename Concat<types::TypeList<LhsTs..., RhsTs...>, Rest...>::Type

10.84.1 Member Typedef Documentation

10.84.1.1 Type

template<typename... LhsTs, typename... RhsTs, typename... Rest>
using serif::utils::misc::concepts::Concat< types::TypeList< LhsTs... >, types::TypeList< RhsTs... >, Rest... >::Type = typename Concat<types::TypeList<LhsTs..., RhsTs...>, Rest...>::Type

The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/concat.hpp

10.85 serif::utils::misc::concepts::Concat< types::TypeList< Ts... > > Struct Template Reference

#include <concat.hpp>

Public Types

- using Type = types::TypeList<Ts...>

10.85.1 Member Typedef Documentation

10.85.1.1 Type

template<typename... Ts>
using serif::utils::misc::concepts::Concat< types::TypeList< Ts... > >::Type = types::TypeList<Ts...>

The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/concat.hpp

10.86 serif::utils::misc::concepts::Concat<> Struct Reference

#include <concat.hpp>

Public Types

- using Type = types::TypeList<>

10.86.1 Member Typedef Documentation

10.86.1.1 Type

using serif::utils::misc::concepts::Concat<>::Type = types::TypeList<>

The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/concat.hpp

10.87 serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics Struct Reference

#include <runtime.hpp>

Public Attributes

- int elementID {-1}
- int domainElementCount {0}
- int visitedElementCount {0}

10.87.1 Member Data Documentation

10.87.1.1 domainElementCount

int serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics::domainElementCount {0}

10.87.1.2 elementID

int serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics::elementID {-1}

10.87.1.3 visitedElementCount

int serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics::visitedElementCount {0}

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/runtime.hpp

10.88 serif::discretization::domain::schema::validation::RelationValidationResult::ConnectedDiagnostics Struct Reference

#include <runtime.hpp>

Public Attributes

- int elementID {-1}
- int domainElementCount {0}
- int visitedElementCount {0}

10.88.1 Member Data Documentation

10.88.1.1 domainElementCount

int serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics::domainElementCount {0}

10.88.1.2 elementID

int serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics::elementID {-1}

10.88.1.3 visitedElementCount

int serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics::visitedElementCount {0}

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/runtime.hpp

10.89 serif::equations::Contribution< TestOperandT, SupportT > Struct Template Reference

Optional authoring base: a test operation and the domains of one weak contribution.

#include <contributions.hpp>

Public Types

- using Test = typename TestOperandT::term_type
- using TestOperand = TestOperandT
- using Support = SupportT

10.89.1 Detailed Description

template<discretization::forms::IsOperand TestOperandT, typename SupportT = discretization::blocks::↩ SupportOf<typename TestOperandT::term_type>>
struct serif::equations::Contribution< TestOperandT, SupportT >

Optional authoring base: a test operation and the domains of one weak contribution.

10.89.2 Member Typedef Documentation

10.89.2.1 Support

template<discretization::forms::IsOperand TestOperandT, typename SupportT = discretization::blocks::SupportOf<typename TestOperandT::term_type>>
using serif::equations::Contribution< TestOperandT, SupportT >::Support = SupportT

10.89.2.2 Test

template<discretization::forms::IsOperand TestOperandT, typename SupportT = discretization::blocks::SupportOf<typename TestOperandT::term_type>>
using serif::equations::Contribution< TestOperandT, SupportT >::Test = typename TestOperandT::term_type

10.89.2.3 TestOperand

template<discretization::forms::IsOperand TestOperandT, typename SupportT = discretization::blocks::SupportOf<typename TestOperandT::term_type>>
using serif::equations::Contribution< TestOperandT, SupportT >::TestOperand = TestOperandT

The documentation for this struct was generated from the following file:

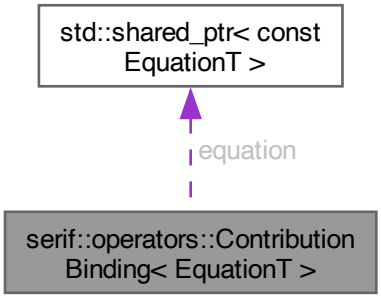
- src/include/serif/equations/contributions.hpp

10.90 serif::operators::ContributionBinding< EquationT > Struct Template Reference

Retain one leaf equation and the constant coefficient of its arithmetic expression.

#include <contributions.hpp>

Collaboration diagram for serif::operators::ContributionBinding< EquationT >:



Public Attributes

- std::shared_ptr< const EquationT > equation
- double coefficient

10.90.1 Detailed Description

template<typename EquationT>
struct serif::operators::ContributionBinding< EquationT >

Retain one leaf equation and the constant coefficient of its arithmetic expression.

10.90.2 Member Data Documentation

10.90.2.1 coefficient

template<typename EquationT>
double serif::operators::ContributionBinding< EquationT >::coefficient

10.90.2.2 equation

template<typename EquationT>
std::shared_ptr<const EquationT> serif::operators::ContributionBinding< EquationT >::equation

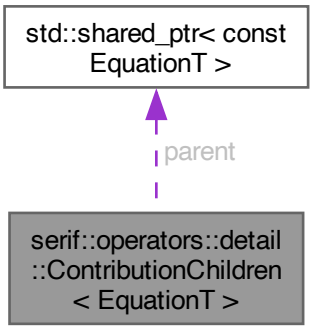
The documentation for this struct was generated from the following file:

- src/include/serif/operators/contributions.hpp

10.91 serif::operators::detail::ContributionChildren< EquationT > Struct Template Reference

#include <contributions.hpp>

Collaboration diagram for serif::operators::detail::ContributionChildren< EquationT >:



Public Member Functions

- ContributionChildren (std::shared_ptr< const EquationT > equation)
  Evaluate contributions once, retaining either its returned value or its returned reference.

Public Attributes

- std::shared_ptr< const EquationT > parent
- decltype(std::declval< const EquationT & >().contributions()) values

10.91.1 Detailed Description

template<typename EquationT>
struct serif::operators::detail::ContributionChildren< EquationT >

A returned tuple may own generated equations or borrow an equation's existing children. Retaining its parent keeps reference tuples valid without copying move-only equations.

10.91.2 Constructor & Destructor Documentation

10.91.2.1 ContributionChildren()

template<typename EquationT>
serif::operators::detail::ContributionChildren< EquationT >::ContributionChildren (
std::shared_ptr< const EquationT > equation) [inline], [explicit]

Evaluate contributions once, retaining either its returned value or its returned reference.

10.91.3 Member Data Documentation

10.91.3.1 parent

template<typename EquationT>
std::shared_ptr<const EquationT> serif::operators::detail::ContributionChildren< EquationT >::parent

10.91.3.2 values

template<typename EquationT>
decltype(std::declval<const EquationT&>().contributions()) serif::operators::detail::ContributionChildren< EquationT >↩
::values

The documentation for this struct was generated from the following file:

- src/include/serif/operators/contributions.hpp

10.92 serif::operators::backend::mfem::ContributionData< OperandCount > Struct Template Reference

#include <volume_data.hpp>

Public Attributes

- ElementBatch< ElementData< OperandCount > > elements
- double residual_offset {0.0}

10.92.1 Member Data Documentation

10.92.1.1 elements

template<std::size_t OperandCount>
ElementBatch<ElementData<OperandCount> > serif::operators::backend::mfem::ContributionData< OperandCount >::elements

10.92.1.2 residual_offset

template<std::size_t OperandCount>
double serif::operators::backend::mfem::ContributionData< OperandCount >::residual_offset {0.0}

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.93 serif::operators::detail::ContributionOperands< TupleT > Struct Template Reference

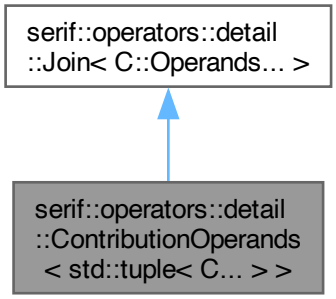
The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

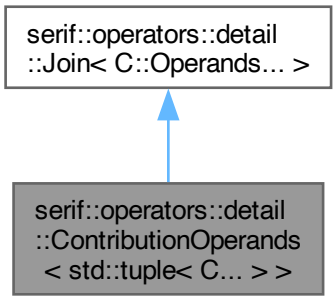
10.94 serif::operators::detail::ContributionOperands< std::tuple< C... > > Struct Template Reference

#include <plan.hpp>

Inheritance diagram for serif::operators::detail::ContributionOperands< std::tuple< C... > >:



Collaboration diagram for serif::operators::detail::ContributionOperands< std::tuple< C... > >:



Additional Inherited Members

Public Types inherited from serif::operators::detail::Join< C::Operands... >

- using type

The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

10.95 serif::equations::Contributions< T, typename > Struct Template Reference

Collect leaf equation types without copying the objects in a composition.

#include <contributions.hpp>

Public Types

- using type = std::tuple<T>

10.95.1 Detailed Description

template<typename T, typename = void>
struct serif::equations::Contributions< T, typename >

Collect leaf equation types without copying the objects in a composition.

10.95.2 Member Typedef Documentation

10.95.2.1 type

template<typename T, typename = void>
using serif::equations::Contributions< T, typename >::type = std::tuple<T>

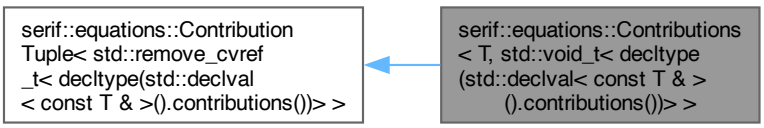
The documentation for this struct was generated from the following file:

- src/include/serif/equations/contributions.hpp

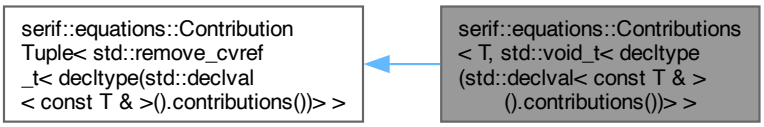
10.96 serif::equations::Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions())> > Struct Template Reference

#include <contributions.hpp>

Inheritance diagram for serif::equations::Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions())> >:



Collaboration diagram for serif::equations::Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions())> >:



Public Types

- using type

Public Types inherited from serif::equations::ContributionTuple< std::remove_cvref_t< decltype(std::declval< const T & >().contrib

- using type

10.96.1 Member Typedef Documentation

10.96.1.1 type

using serif::equations::Contributions< T, typename >::type

The documentation for this struct was generated from the following file:

- src/include/serif/equations/contributions.hpp

10.97 serif::equations::ContributionSupport< T, typename > Struct Template Reference

#include <contributions.hpp>

Public Types

- using type = discretization::blocks::SupportOf<typename T::Test>

10.97.1 Member Typedef Documentation

10.97.1.1 type

template<typename T, typename = void>
using serif::equations::ContributionSupport< T, typename >::type = discretization::blocks::SupportOf<typename T::Test>

The documentation for this struct was generated from the following file:

- src/include/serif/equations/contributions.hpp

10.98 serif::equations::ContributionSupport< T, std::void_t< typename T::Support > > Struct Template Reference

#include <contributions.hpp>

Public Types

- using type = typename T::Support
- using type

10.98.1 Member Typedef Documentation

10.98.1.1 type [1/2]

using serif::equations::ContributionSupport< T, typename >::type

10.98.1.2 type [2/2]

template<typename T>
using serif::equations::ContributionSupport< T, std::void_t< typename T::Support > >::type = typename T::Support

The documentation for this struct was generated from the following file:

- src/include/serif/equations/contributions.hpp

10.99 serif::equations::ContributionTuple< TupleT, typename > Struct Template Reference

#include <contributions.hpp>

Public Types

- using type = void

10.99.1 Member Typedef Documentation

10.99.1.1 type

template<typename TupleT, typename = void>
using serif::equations::ContributionTuple< TupleT, typename >::type = void

The documentation for this struct was generated from the following file:

- src/include/serif/equations/contributions.hpp

10.100 serif::equations::ContributionTuple< std::tuple< Ts... >, std::void_t< decltype(std::tuple_cat(std::declval< typename Contributions< std::remove_cvref_t< Ts > >::type >()...))> > Struct Template Reference

#include <contributions.hpp>

Public Types

- using type
- using type

10.100.1 Member Typedef Documentation

10.100.1.1 type [1/2]

using serif::equations::ContributionTuple< std::tuple< Ts... >, typename >::type

10.100.1.2 type [2/2]

template<typename... Ts>
using serif::equations::ContributionTuple< std::tuple< Ts... >, std::void_t< decltype(std::tuple_cat(std::declval< typename Contributions< std::remove_cvref_t< Ts > >::type >()...))> >::type

Initial value:

```
decltype(std::tuple_cat(
        std::declval<typename Contributions<std::remove_cvref_t<Ts>::type>()...
    ))
```

The documentation for this struct was generated from the following file:

- src/include/serif/equations/contributions.hpp

10.101 serif::utils::dependencies::keys::Coordinates Struct Reference

Positions supplied to a computation, in the caller's coordinate convention.

#include <keys.hpp>

10.101.1 Detailed Description

Positions supplied to a computation, in the caller's coordinate convention.

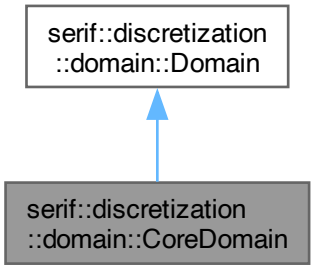
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/keys.hpp

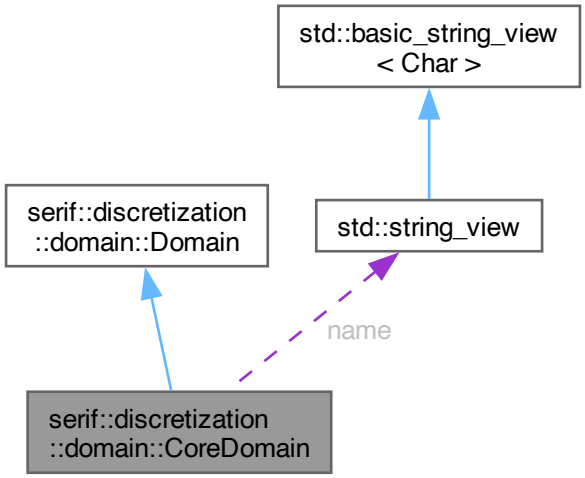
10.102 serif::discretization::domain::CoreDomain Struct Reference

#include <types.hpp>

Inheritance diagram for serif::discretization::domain::CoreDomain:



Collaboration diagram for serif::discretization::domain::CoreDomain:



Static Public Attributes

- static constexpr std::string_view name = ”core”

10.102.1 Member Data Documentation

10.102.1.1 name

std::string_view serif::discretization::domain::CoreDomain::name = ”core” [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/types.hpp

10.103 serif::operators::detail::CouplingForm< Input, TestOperand > Struct Template Reference

#include <plan.hpp>

Public Types

- using Trial = typename Input::term_type
- using Test = typename TestOperand::term_type
- using type

10.103.1 Member Typedef Documentation

10.103.1.1 Test

```
template<typename Input, typename TestOperand>
using serif::operators::detail::CouplingForm< Input, TestOperand >::Test = typename TestOperand::term_type
```

10.103.1.2 Trial

```
template<typename Input, typename TestOperand>
using serif::operators::detail::CouplingForm< Input, TestOperand >::Trial = typename Input::term_type
```

10.103.1.3 type

```
template<typename Input, typename TestOperand>
using serif::operators::detail::CouplingForm< Input, TestOperand >::type
```

Initial value:

```
std::conditional_t<
        std::same_as<typename Input::operation_type, discretization::forms::Operation::Divergence>,
        discretization::forms::divergence<Trial, Test>,
        std::conditional_t<std::same_as<typename TestOperand::operation_type, discretization::forms::Operation::Divergence>,
                discretization::forms::divergence<Test, Trial>,
                std::conditional_t<std::same_as<Trial, Test>, discretization::forms::mass<Trial, Test>,
                        discretization::forms::source<Trial, Test>>>
```

The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

10.104 serif::utils::types::ct_error_string< N > Struct Template Reference

#include <strings.hpp>

Public Member Functions

- constexpr ct_error_string (const char(&str)[N])
- constexpr std::string_view view () const

Public Attributes

- char data [289+N] {}
- char padding [289] {"NOTE: This error message is generated at compile time. C++ < 26 limits how compile time errors can be formatted, This is the relevant error message not downstream warnings about implicit instantiations of an undefined template. Those are used to force the compiler to display this message"}

10.104.1 Constructor & Destructor Documentation

10.104.1.1 ct_error_string()

template<std::size_t N>
serif::utils::types::ct_error_string< N >::ct_error_string (
const char(&) str[N]) [inline], [constexpr]

10.104.2 Member Function Documentation

10.104.2.1 view()

template<std::size_t N>
std::string_view serif::utils::types::ct_error_string< N >::view () const [inline], [constexpr]

10.104.3 Member Data Documentation

10.104.3.1 data

template<std::size_t N>
char serif::utils::types::ct_error_string< N >::data[289+N] {}

10.104.3.2 padding

template<std::size_t N>
char serif::utils::types::ct_error_string< N >::padding[289] {"NOTE: This error message is generated at compile time. C++ < 26 limits how compile time errors can be formatted, This is the relevant error message not downstream warnings about implicit instantiations of an undefined template. Those are used to force the compiler to display this message"}

The documentation for this struct was generated from the following file:

- src/include/serif/utils/types/strings.hpp

10.105 serif::utils::types::ctstring< N > Struct Template Reference

#include <strings.hpp>

Public Member Functions

- constexpr ctstring (const char(&str)[N])
- constexpr std::string_view view () const

Public Attributes

- char data [N] {}

10.105.1 Constructor & Destructor Documentation

10.105.1.1 ctstring()

template<std::size_t N>
serif::utils::types::ctstring< N >::ctstring (
const char(&) str[N]) [inline], [constexpr]

10.105.2 Member Function Documentation

10.105.2.1 view()

template<std::size_t N>
std::string_view serif::utils::types::ctstring< N >::view () const [inline], [constexpr]

10.105.3 Member Data Documentation

10.105.3.1 data

template<std::size_t N>
char serif::utils::types::ctstring< N >::data[N] {}

The documentation for this struct was generated from the following file:

- src/include/serif/utils/types/strings.hpp

10.106 serif::discretization::forms::Operation::Curl Struct Reference

Operation tag for evaluating a curl.

#include <operands.hpp>

10.106.1 Detailed Description

Operation tag for evaluating a curl.

The documentation for this struct was generated from the following file:

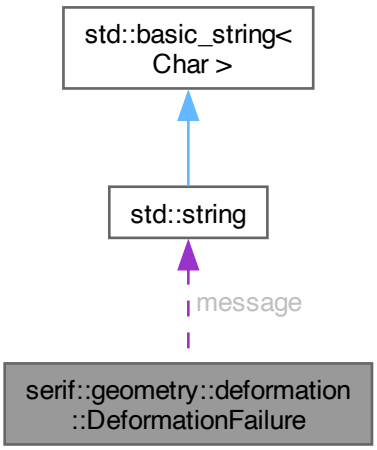
- src/include/serif/discretization/forms/operands.hpp

10.107 serif::geometry::deformation::DeformationFailure Struct Reference

Describe a rejected geometry calculation; the dependency cache publishes results only after success.

#include <failure.hpp>

Collaboration diagram for serif::geometry::deformation::DeformationFailure:



Public Attributes

- DeformationFailureReason reason
- std::string message
- int element {-1}
- int point {-1}

10.107.1 Detailed Description

Describe a rejected geometry calculation; the dependency cache publishes results only after success.

10.107.2 Member Data Documentation

10.107.2.1 element

int serif::geometry::deformation::DeformationFailure::element {-1}

10.107.2.2 message

std::string serif::geometry::deformation::DeformationFailure::message

10.107.2.3 point

int serif::geometry::deformation::DeformationFailure::point {-1}

10.107.2.4 reason

DeformationFailureReason serif::geometry::deformation::DeformationFailure::reason

The documentation for this struct was generated from the following file:

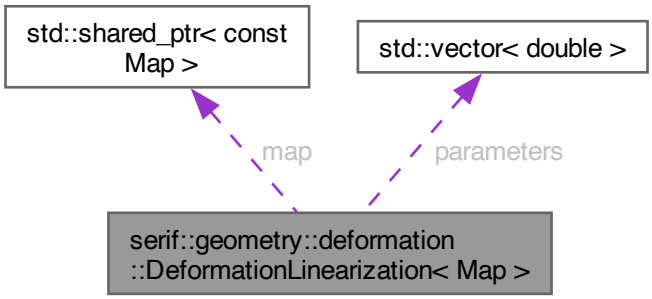
- src/include/serif/geometry/deformation/failure.hpp

10.108 serif::geometry::deformation::DeformationLinearization< Map > Struct Template Reference

Freeze the shape coordinates associated with a derivative, independently of mutable caller inputs.

#include <linearization.hpp>

Collaboration diagram for serif::geometry::deformation::DeformationLinearization< Map >:



Public Attributes

- std::shared_ptr< const Map > map
- std::vector< double > parameters

10.108.1 Detailed Description

template<DeformationMap Map>
struct serif::geometry::deformation::DeformationLinearization< Map >

Freeze the shape coordinates associated with a derivative, independently of mutable caller inputs.

10.108.2 Member Data Documentation

10.108.2.1 map

template<DeformationMap Map>
std::shared_ptr<const Map> serif::geometry::deformation::DeformationLinearization< Map >::map

10.108.2.2 parameters

template<DeformationMap Map>
std::vector<double> serif::geometry::deformation::DeformationLinearization< Map >::parameters

The documentation for this struct was generated from the following file:

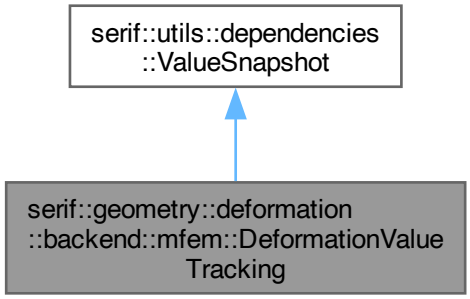
- src/include/serif/geometry/deformation/linearization.hpp

10.109 serif::geometry::deformation::backend::mfem::DeformationValue↩ Tracking Struct Reference

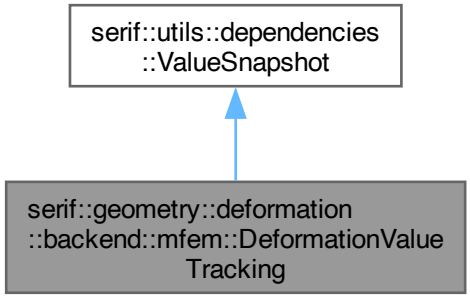
Model options and coefficient copies describe values independently of wrapper addresses.

#include <inputs.hpp>

Inheritance diagram for serif::geometry::deformation::backend::mfem::DeformationValueTracking:



Collaboration diagram for serif::geometry::deformation::backend::mfem::DeformationValueTracking:



Public Member Functions

- template<typename Value>
  const void * identity (const Value &) const noexcept

  Equal physical values remain reusable when a context is reconstructed.

Public Member Functions inherited from serif::utils::dependencies::ValueSnapshot

- template<typename T>
  requires std::copy_constructible<T> && std::equality_comparable<T>
  T capture (const T &value) const

Keep the accepted value independent of subsequent edits to a borrowed source.

- template<typename T>
  requires std::equality_comparable<T>
  bool matches (const T &captured, const T &value) const

  Compare content rather than a hash that could silently miss a change.

10.109.1 Detailed Description

Model options and coefficient copies describe values independently of wrapper addresses.

10.109.2 Member Function Documentation

10.109.2.1 identity()

template<typename Value>
const void * serif::geometry::deformation::backend::mfem::DeformationValueTracking::identity (
const Value &) const [inline], [nodiscard], [noexcept]

Equal physical values remain reusable when a context is reconstructed.

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/backend/mfem/inputs.hpp

10.110 serif::geometry::deformation::DeformationWorkspaceType< Map, Linear > Struct Template Reference

Select numerical workspace storage without adding a preparation owner.

#include <actions.hpp>

Public Types

- using Type = typename Map::Workspace

10.110.1 Detailed Description

template<DeformationMap Map, bool Linear = LinearDeformationMap<Map>>
struct serif::geometry::deformation::DeformationWorkspaceType< Map, Linear >

Select numerical workspace storage without adding a preparation owner.

10.110.2 Member Typedef Documentation

10.110.2.1 Type

template<DeformationMap Map, bool Linear = LinearDeformationMap<Map>>
using serif::geometry::deformation::DeformationWorkspaceType< Map, Linear >::Type = typename Map::Workspace

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/actions.hpp

10.111 serif::geometry::deformation::DeformationWorkspaceType< Map, true > Struct Template Reference

#include <actions.hpp>

Public Types

- using Type = EmptyDeformationWorkspace
- using Type

10.111.1 Member Typedef Documentation

10.111.1.1 Type [1/2]

using serif::geometry::deformation::DeformationWorkspaceType< Map, Linear >::Type

10.111.1.2 Type [2/2]

template<DeformationMap Map>
using serif::geometry::deformation::DeformationWorkspaceType< Map, true >::Type = EmptyDeformationWorkspace

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/actions.hpp

10.112 serif::utils::types::DenseMatrixBase< T > Struct Template Reference

#include <tensor.hpp>

Public Types

- using Type = T

10.112.1 Member Typedef Documentation

10.112.1.1 Type

template<typename T>
using serif::utils::types::DenseMatrixBase< T >::Type = T

The documentation for this struct was generated from the following file:

- src/include/serif/utils/types/tensor/tensor.hpp

10.113 serif::utils::types::DenseTensorBase< T, Rank > Struct Template Reference

#include <tensor.hpp>

Public Types

- using Type = T

Static Public Attributes

- static constexpr std::size_t rank = Rank

10.113.1 Member Typedef Documentation

10.113.1.1 Type

template<typename T, std::size_t Rank>
using serif::utils::types::DenseTensorBase< T, Rank >::Type = T

10.113.2 Member Data Documentation

10.113.2.1 rank

template<typename T, std::size_t Rank>
std::size_t serif::utils::types::DenseTensorBase< T, Rank >::rank = Rank [static], [constexpr]

The documentation for this struct was generated from the following file:

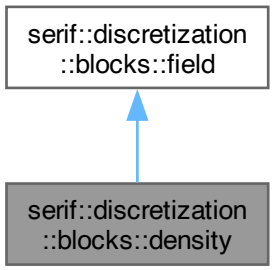
- src/include/serif/utils/types/tensor/tensor.hpp

10.114 serif::discretization::blocks::density Struct Reference

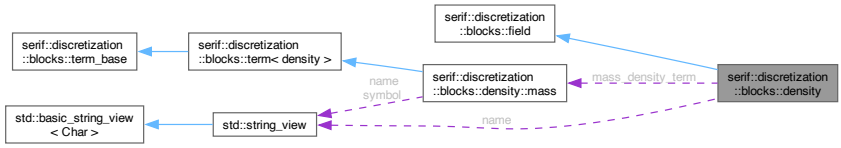
The density field used to represent the distribution of mass in the system. This is a scalar field that is associated with the mass density of the system. The density field is used to enforce the mass conservation condition in the weak form.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::density:



Collaboration diagram for serif::discretization::blocks::density:



Classes

- struct mass

  The mass density term for the system. Note that the naming here could be a bit confusing as FEM also uses the term ”mass” to represent the canonical mass matrix. This is a more standard physics reading of what is going on here. We define that this space is a second order scalar L2 space. This allows for the possibility of discontinuous density fields.

Public Types

- using Terms = utils::types::TypeList<mass>

  Type list of terms for compile time reflection and iteration over the terms of the field.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<density>()

  Compile time reflected name of the field.
- static constexpr mass mass_density_term {}

  Static constexpr instance of the mass term. This is used to access the term in the discretization.

10.114.1 Detailed Description

The density field used to represent the distribution of mass in the system. This is a scalar field that is associated with the mass density of the system. The density field is used to enforce the mass conservation condition in the weak form.

10.114.2 Member Typedef Documentation

10.114.2.1 Terms

using serif::discretization::blocks::density::Terms = utils::types::TypeList<mass>

Type list of terms for compile time reflection and iteration over the terms of the field.

10.114.3 Member Data Documentation

10.114.3.1 mass_density_term

mass serif::discretization::blocks::density::mass_density_term {} [static], [constexpr]

Static constexpr instance of the mass term. This is used to access the term in the discretization.

10.114.3.2 name

std::string_view serif::discretization::blocks::density::name = meta::type_name<density>() [static], [constexpr]

Compile time reflected name of the field.

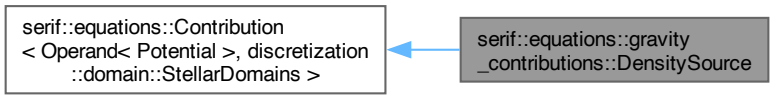
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

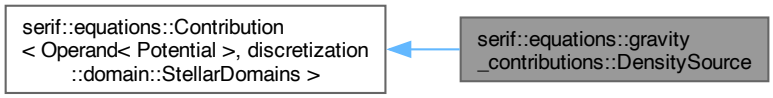
10.115 serif::equations::gravity_contributions::DensitySource Struct Reference

#include <mixed_gravity.hpp>

Inheritance diagram for serif::equations::gravity_contributions::DensitySource:



Collaboration diagram for serif::equations::gravity_contributions::DensitySource:



Public Types

- using Operands = utils::types::TypeList<Operand<Density>>

Public Types inherited from serif::equations::Contribution< Operand< Potential >, discretization::domain::StellarDomains >

- using Test
- using TestOperand
- using Support

Public Member Functions

- DensitySource (double coefficient)
- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const

  Only stellar elements contribute -4 pi G q rho to the Poisson volume row.
- int additional_order (Operand< Density >, int) const

Public Attributes

- double four_pi_G

10.115.1 Member Typedef Documentation

10.115.1.1 Operands

using serif::equations::gravity_contributions::DensitySource::Operands = utils::types::TypeList<Operand<Density>>

10.115.2 Constructor & Destructor Documentation

10.115.2.1 DensitySource()

serif::equations::gravity_contributions::DensitySource::DensitySource (
double coefficient) [inline], [explicit]

10.115.3 Member Function Documentation

10.115.3.1 additional_order()

int serif::equations::gravity_contributions::DensitySource::additional_order (
Operand< Density > ,
int) const [inline]

10.115.3.2 evaluate()

PointLinearization< Operands > serif::equations::gravity_contributions::DensitySource::evaluate (
const EquationPoint< Operands > & point) const [inline]

Only stellar elements contribute -4 pi G q rho to the Poisson volume row.

10.115.4 Member Data Documentation

10.115.4.1 four_pi_G

double serif::equations::gravity_contributions::DensitySource::four_pi_G

The documentation for this struct was generated from the following file:

- src/include/serif/equations/mixed_gravity.hpp

10.116 serif::utils::dependencies::DependencyContext< ResourceTs > Class Template Reference

#include <dependency_context.hpp>

Public Types

- using Resources = types::TypeList<ResourceTs...>
- using Bindings = dependencies::Bindings<Binding<typename ResourceTs::Key, typename ResourceTs::Target>...>
- using ResourceBindings = dependencies::Bindings<Binding<typename ResourceTs::Key, ResourceTs>...>

Public Member Functions

- DependencyContext (ResourceTs... resources)

  Each resource explicitly chooses owned or borrowed storage.
- template<typename KeyT>
  requires BindingsWithBinding<Bindings, KeyT>
  const auto & get () const

  Retrieve a resource by role even when several roles have the same target type.
- template<typename KeyT>
  requires BindingsWithBinding<Bindings, KeyT>
  const auto & resource () const

  Access the value together with the policy used to track it.
- template<IsKeyList KeysT>
  requires SatisfiesKeyList<Bindings, KeysT>
  auto select (KeysT) const &

  Select declared roles without copying their objects or tracking policies.
- template<IsKeyList KeysT>
  auto select (KeysT) const &&=delete

  A view cannot safely borrow a temporary context.

Private Attributes

- std::tuple< ResourceTs... > m_resources

10.116.1 Member Typedef Documentation

10.116.1.1 Bindings

template<IsResource... ResourceTs>
using serif::utils::dependencies::DependencyContext< ResourceTs >::Bindings = dependencies::Bindings<Binding<typename ResourceTs::Key, typename ResourceTs::Target>...>

10.116.1.2 ResourceBindings

template<IsResource... ResourceTs>
using serif::utils::dependencies::DependencyContext< ResourceTs >::ResourceBindings = dependencies::Bindings<Binding<typename ResourceTs::Key, ResourceTs>...>

10.116.1.3 Resources

template<IsResource... ResourceTs>
using serif::utils::dependencies::DependencyContext< ResourceTs >::Resources = types::TypeList<ResourceTs...>

10.116.2 Constructor & Destructor Documentation

10.116.2.1 DependencyContext()

template<IsResource... ResourceTs>
serif::utils::dependencies::DependencyContext< ResourceTs >::DependencyContext (
ResourceTs... resources) [inline], [explicit]

Each resource explicitly chooses owned or borrowed storage.

10.116.3 Member Function Documentation

10.116.3.1 get()

template<IsResource... ResourceTs>
template<typename KeyT>
requires BindingsWithBinding<Bindings, KeyT>
const auto & serif::utils::dependencies::DependencyContext< ResourceTs >::get () const [inline], [nodiscard]

Retrieve a resource by role even when several roles have the same target type.

10.116.3.2 resource()

template<IsResource... ResourceTs>
template<typename KeyT>
requires BindingsWithBinding<Bindings, KeyT>
const auto & serif::utils::dependencies::DependencyContext< ResourceTs >::resource () const [inline], [nodiscard]

Access the value together with the policy used to track it.

10.116.3.3 select() [1/2]

```
template<IsResource... ResourceTs>
template<IsKeyList KeysT>
requires SatisfiesKeyList<Bindings, KeysT>
auto serif::utils::dependencies::DependencyContext< ResourceTs >::select (
                KeysT ) const &    [inline], [nodiscard]
```

Select declared roles without copying their objects or tracking policies.

10.116.3.4 select() [2/2]

```
template<IsResource... ResourceTs>
template<IsKeyList KeysT>
auto serif::utils::dependencies::DependencyContext< ResourceTs >::select (
                KeysT ) const &&    [delete]
```

A view cannot safely borrow a temporary context.

10.116.4 Member Data Documentation

10.116.4.1 m_resources

```
template<IsResource... ResourceTs>
std::tuple<ResourceTs...> serif::utils::dependencies::DependencyContext< ResourceTs >::m_resources    [private]
```

The documentation for this class was generated from the following file:

- src/include/serif/utils/dependencies/dependency_context.hpp

10.117 serif::utils::dependencies::DependencyContextView< ParentT, KeysT > Class Template Reference

The documentation for this class was generated from the following file:

- src/include/serif/utils/dependencies/dependency_context.hpp

10.118 serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > > Class Template Reference

#include <dependency_context.hpp>

Public Types

- using Bindings = dependencies::Bindings<Binding<Keys, bound_type_t<typename ParentT::↩ Bindings, Keys>>...>
- using ResourceBindings = dependencies::Bindings<Binding<Keys, bound_type_t<typename Parent↩ T::ResourceBindings, Keys>>...>
- using Resources = types::TypeList<bound_type_t<typename ParentT::ResourceBindings, Keys>...>

Public Member Functions

- DependencyContextView (const ParentT &parent)

  The parent context must outlive this view.
- DependencyContextView (const ParentT &&)=delete

  Reject temporary parents even when constructing a view directly.
- template<typename KeyT>
  requires BindingsWithBinding<Bindings, KeyT>
  const auto & get () const

  Expose only the declared portion of the parent context.
- template<typename KeyT>
  requires BindingsWithBinding<Bindings, KeyT>
  const auto & resource () const

  Preserve the original source identity and tracking policy through selection.

Private Attributes

- const ParentT & m_parent

10.118.1 Member Typedef Documentation

10.118.1.1 Bindings

template<typename ParentT, typename... Keys>
using serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >::Bindings = dependencies::Bindings<Binding<Keys, bound_type_t<typename ParentT::Bindings, Keys>>...>

10.118.1.2 ResourceBindings

template<typename ParentT, typename... Keys>
using serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >::ResourceBindings = dependencies::Bindings<Binding<K bound_type_t<typename ParentT::ResourceBindings, Keys>>...>

10.118.1.3 Resources

template<typename ParentT, typename... Keys>
using serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >::Resources = types::TypeList<bound_type_t<typename ParentT::ResourceBindings, Keys>...>

10.118.2 Constructor & Destructor Documentation

10.118.2.1 DependencyContextView() [1/2]

template<typename ParentT, typename... Keys>
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >::DependencyContextView (
const ParentT & parent) [inline], [explicit]

The parent context must outlive this view.

10.118.2.2 DependencyContextView() [2/2]

template<typename ParentT, typename... Keys>
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >::DependencyContextView (
const ParentT &&) [delete]

Reject temporary parents even when constructing a view directly.

10.118.3 Member Function Documentation

10.118.3.1 get()

template<typename ParentT, typename... Keys>
template<typename KeyT>
requires BindingsWithBinding<Bindings, KeyT>
const auto & serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >::get () const [inline], [nodiscard]

Expose only the declared portion of the parent context.

10.118.3.2 resource()

template<typename ParentT, typename... Keys>
template<typename KeyT>
requires BindingsWithBinding<Bindings, KeyT>
const auto & serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >::resource () const [inline], [nodiscard]

Preserve the original source identity and tracking policy through selection.

10.118.4 Member Data Documentation

10.118.4.1 m_parent

template<typename ParentT, typename... Keys>
const ParentT& serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >::m_parent [private]

The documentation for this class was generated from the following file:

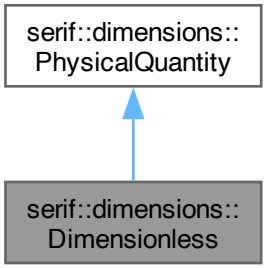
- src/include/serif/utils/dependencies/dependency_context.hpp

10.119 serif::dimensions::Dimensionless Struct Reference

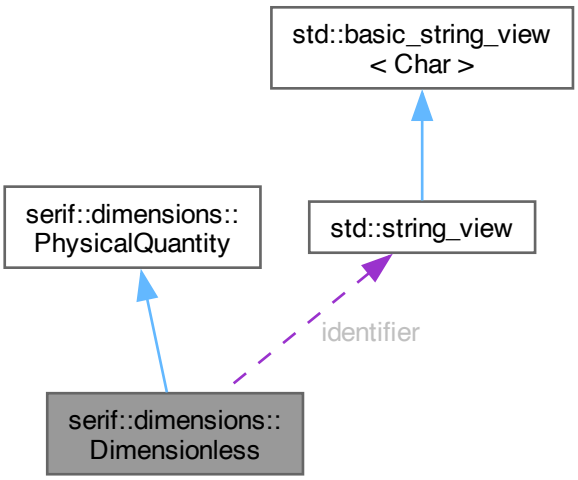
Non dimensional type.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Dimensionless:



Collaboration diagram for serif::dimensions::Dimensionless:



Static Public Attributes

- static constexpr std::string_view identifier = "dimensionless"

10.119.1 Detailed Description

Non dimensional type.

10.119.2 Member Data Documentation

10.119.2.1 identifier

std::string_view serif::dimensions::Dimensionless::identifier = "dimensionless" [static], [constexpr]

The documentation for this struct was generated from the following file:

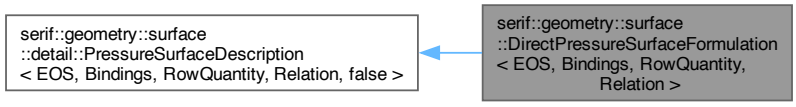
- src/include/serif/dimensions/dimensions.hpp

10.120 serif::geometry::surface::DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation > Struct Template Reference

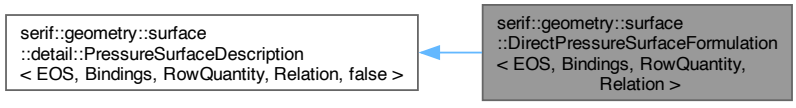
Enforce P_EOS(state) - P_surface = 0 without inventing an inverse EOS relation.

#include <compilation.hpp>

Inheritance diagram for serif::geometry::surface::DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation >:



Collaboration diagram for serif::geometry::surface::DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation >:



Additional Inherited Members

Public Types inherited from serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, Relation, false

- using EquationOfState
- using StateBindings
- using Relation
- using PhysicalCondition
- using PhysicalQuantity
- using ResidualQuantity
- using CarrierQuantity
- using CarrierTerm
- using Test
- using Support
- using EOSStateQuantities
- using StateQuantities
- using StateTerms
- using EOSStateTerms
- using Operands

Static Public Attributes inherited from serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, Relation, false

- static constexpr bool resolves_carrier

10.120.1 Detailed Description

template<typename EOS, typename Bindings, typename RowQuantity, typename Relation>
requires DirectPressureSurfaceCompilable<EOS, Bindings, RowQuantity, Relation>
struct serif::geometry::surface::DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation >

Enforce P_EOS(state) - P_surface = 0 without inventing an inverse EOS relation.

Note

The explicit row and relation make the choice of residual, units, and Newton scaling visible.

The documentation for this struct was generated from the following file:

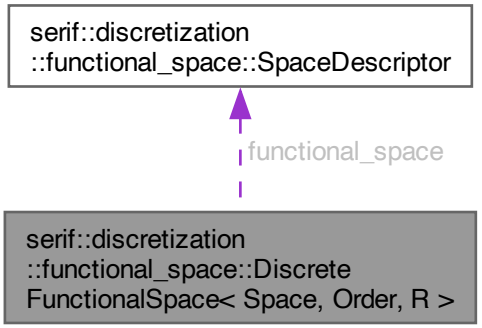
- src/include/serif/geometry/surface/compilation.hpp

10.121 serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > Struct Template Reference

Compile-time descriptor for a discrete functional space. The difference between a FunctionalSpace and a discrete functional space is that a discrete functional space carries some additional information which is needed to actually construct a finite element space (e.g. polynomial order, and rank). Once you have a discrete functional space you have, for MFEM at least, enough information to fully define a finite element space.

#include <space.hpp>

Collaboration diagram for serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >:



Public Types

- using rank = Rank<R>

Static Public Attributes

- static constexpr SpaceDescriptor functional_space = Space
- static constexpr int order = Order

10.121.1 Detailed Description

template<SpaceDescriptor Space, int Order, std::size_t R>
struct serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >

Compile-time descriptor for a discrete functional space. The difference between a FunctionalSpace and a discrete functional space is that a discrete functional space carries some additional information which is needed to actually construct a finite element space (e.g. polynomial order, and rank). Once you have a discrete functional space you have, for MFEM at least, enough information to fully define a finite element space.

Template Parameters

Space	The functional space descriptor (e.g. L2, H1, H(div), H(curl)).
Order	The order of the finite element family (e.g. 0 for P0, 1 for P1, etc.).
R	The rank of the functional space (e.g. 0 for scalar, 1 for vector, 2 for tensor).

10.121.2 Member Typedef Documentation

10.121.2.1 rank

template<SpaceDescriptor Space, int Order, std::size_t R>
using serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >::rank = Rank<R>

10.121.3 Member Data Documentation

10.121.3.1 functional_space

template<SpaceDescriptor Space, int Order, std::size_t R>
SpaceDescriptor serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >::functional_space = Space [static], [constexpr]

10.121.3.2 order

template<SpaceDescriptor Space, int Order, std::size_t R>
int serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >::order = Order [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/functional_space/space.hpp

10.122 serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< FunctionalSpaceT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/backend/order.hpp

10.123 serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > > Struct Template Reference

#include <order.hpp>

Static Public Attributes

- static constexpr auto space = SpaceV
- static constexpr auto order = static_cast<int>(OrderV)
- static constexpr std::size_t rank = static_cast<std::size_t>(RankV)

10.123.1 Member Data Documentation

10.123.1.1 order

template<auto SpaceV, auto OrderV, auto RankV>
auto serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > >::order = static_cast<int>(OrderV) [static], [constexpr]

10.123.1.2 rank

template<auto SpaceV, auto OrderV, auto RankV>
std::size_t serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > >::rank = static_cast<std::size_t>(RankV) [static], [constexpr]

10.123.1.3 space

template<auto SpaceV, auto OrderV, auto RankV>
auto serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > >::space = SpaceV [static], [constexpr]

The documentation for this struct was generated from the following file:

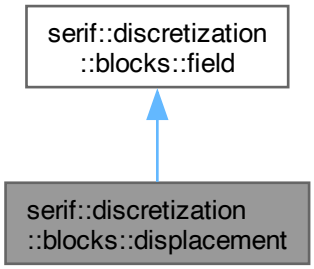
- src/include/serif/discretization/quadrature/backend/order.hpp

10.124 serif::discretization::blocks::displacement Struct Reference

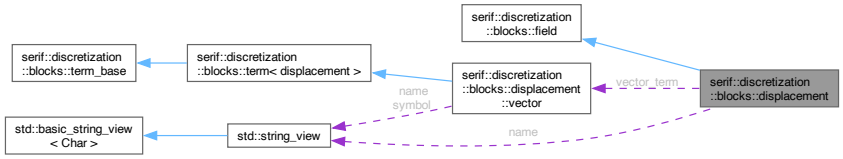
The displacement field is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::displacement:



Collaboration diagram for serif::discretization::blocks::displacement:



Classes

- struct vector

  the displacement vector term is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain. This is a vector field that is associated with the mapping of the domain from the reference to physical spaces. E.g. if there is some domain O_ref which n nodal positions x^i_ref then there is some other domain O_phys with n nodel positions such that x^i_phys = x^i_ref + d^i where d^i is the displacement vector at node i.

Public Types

- using Terms = utils::types::TypeList<vector>

  Type list of terms for compile time reflection and iteration over the terms of the field.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<displacement>()
  Compile time reflected name of the field.
- static constexpr vector vector_term {}
  Static constexpr instance of the vector term. This is used to access the term in the discretization.

10.124.1 Detailed Description

The displacement field is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain.

10.124.2 Member Typedef Documentation

10.124.2.1 Terms

using serif::discretization::blocks::displacement::Terms = utils::types::TypeList<vector>

Type list of terms for compile time reflection and iteration over the terms of the field.

10.124.3 Member Data Documentation

10.124.3.1 name

std::string_view serif::discretization::blocks::displacement::name = meta::type_name<displacement>() [static], [constexpr]

Compile time reflected name of the field.

10.124.3.2 vector_term

vector serif::discretization::blocks::displacement::vector_term {} [static], [constexpr]

Static constexpr instance of the vector term. This is used to access the term in the discretization.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

10.125 serif::operators::backend::mfem::DisplacementCoefficients Struct Reference

#include <volume_inputs.hpp>

The documentation for this struct was generated from the following file:

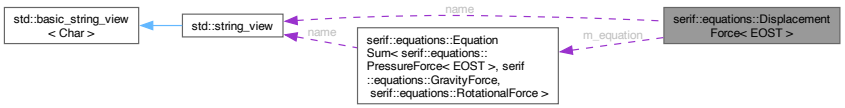
- src/include/serif/operators/backend/mfem/volume_inputs.hpp

10.126 serif::equations::DisplacementForce< EOST > Class Template Reference

Mechanical balance: pressure force + gravity force + rotational force.

#include <displacement_force.hpp>

Collaboration diagram for serif::equations::DisplacementForce< EOST >:



Public Types

- using Test = discretization::blocks::displacement::vector

Public Member Functions

- DisplacementForce (PressureForce< EOST > pressure, RigidRotation::Vector axis={0.0, 0.0, 1.0}, RigidRotation::Vector center={})

  Pressure carries its EOS; rotation prescribes an axis and reads angular speed from the state.
- const auto & contributions () const noexcept

Static Public Attributes

- static constexpr std::string_view name = "displacement force"

Private Attributes

- EquationSum< PressureForce< EOST >, GravityForce, RotationalForce > m_equation

10.126.1 Detailed Description

template<eos::models::PressureForceEOS EOST>
class serif::equations::DisplacementForce< EOST >

Mechanical balance: pressure force + gravity force + rotational force.

10.126.2 Member Typedef Documentation

10.126.2.1 Test

template<eos::models::PressureForceEOS EOST>
using serif::equations::DisplacementForce< EOST >::Test = discretization::blocks::displacement::vector

10.126.3 Constructor & Destructor Documentation

10.126.3.1 DisplacementForce()

template<eos::models::PressureForceEOS EOST>
serif::equations::DisplacementForce< EOST >::DisplacementForce (
PressureForce< EOST > pressure,
RigidRotation::Vector axis = {0.0, 0.0, 1.0},
RigidRotation::Vector center = {}) [inline], [explicit]

Pressure carries its EOS; rotation prescribes an axis and reads angular speed from the state.

10.126.4 Member Function Documentation

10.126.4.1 contributions()

template<eos::models::PressureForceEOS EOST>
const auto & serif::equations::DisplacementForce< EOST >::contributions () const [inline], [nodiscard], [noexcept]

10.126.5 Member Data Documentation

10.126.5.1 m_equation

template<eos::models::PressureForceEOS EOST>
EquationSum<PressureForce<EOST>, GravityForce, RotationalForce> serif::equations::DisplacementForce< EOST >::m_↩
equation [private]

10.126.5.2 name

template<eos::models::PressureForceEOS EOST>
std::string_view serif::equations::DisplacementForce< EOST >::name = "displacement force" [static], [constexpr]

The documentation for this class was generated from the following file:

- src/include/serif/equations/displacement_force.hpp

10.127 serif::discretization::forms::Operation::Divergence Struct Reference

Operation tag for evaluating a divergence.

#include <operands.hpp>

10.127.1 Detailed Description

Operation tag for evaluating a divergence.

The documentation for this struct was generated from the following file:

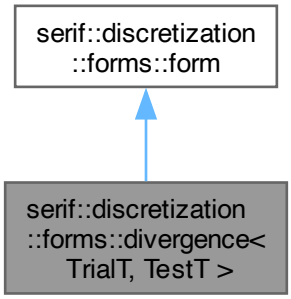
- src/include/serif/discretization/forms/operands.hpp

10.128 serif::discretization::forms::divergence< TrialT, TestT > Struct Template Reference

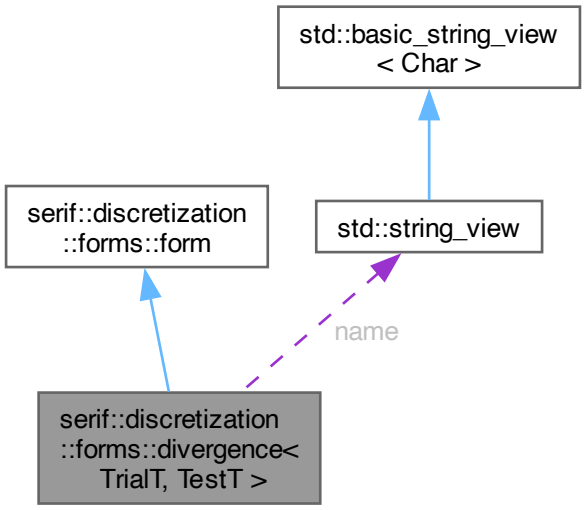
The divergence form represents the divergence operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the divergence operation.

#include <forms.hpp>

Inheritance diagram for serif::discretization::forms::divergence< TrialT, TestT >:



Collaboration diagram for serif::discretization::forms::divergence< TrialT, TestT >:



Public Types

- using trial_type = TrialT
  The trial term type for the divergence form.
- using test_type = TestT
  The test term type for the divergence form.
- using operand_types = utils::types::TypeList<Operand<TrialT, Operation::Divergence>, Operand<TestT>>
  Type list of operand types for the divergence form, used for compile-time reflection and iteration over the operands.

Static Public Attributes

- static constexpr std::string_view name = meta::template_name<divergence>()
  Compile time reflected name of the form.

10.128.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::divergence< TrialT, TestT >

The divergence form represents the divergence operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the divergence operation.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

Note

The divergence form has an order reduction of 1 for the trial term to satisfy the Inf-Sup condition, while the test term has no order reduction. You can see where this reduction is defined in code by looking at the form_traits.hpp file (Specialization for divergence).

10.128.2 Member Typedef Documentation

10.128.2.1 operand_types

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::divergence< TrialT, TestT >::operand_types = utils::types::TypeList<Operand<TrialT, Operation::Divergence>, Operand<TestT>>

Type list of operand types for the divergence form, used for compile-time reflection and iteration over the operands.

10.128.2.2 test_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::divergence< TrialT, TestT >::test_type = TestT

The test term type for the divergence form.

10.128.2.3 trial_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::divergence< TrialT, TestT >::trial_type = TrialT

The trial term type for the divergence form.

10.128.3 Member Data Documentation

10.128.3.1 name

template<blocks::Term TrialT, blocks::Term TestT>
std::string_view serif::discretization::forms::divergence< TrialT, TestT >::name = meta::template_name<divergence>() [static], [constexpr]

Compile time reflected name of the form.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/forms.hpp

10.129 serif::mapping::DMFixedMeshGeometry Struct Reference

#include <data.hpp>

Public Attributes

- mfem::Vector position
- mfem::DenseMatrix jacobian
- mfem::DenseMatrix inverse_jacobian
- double weight {0.0}

10.129.1 Member Data Documentation

10.129.1.1 inverse_jacobian

mfem::DenseMatrix serif::mapping::DMFixedMeshGeometry::inverse_jacobian

10.129.1.2 jacobian

mfem::DenseMatrix serif::mapping::DMFixedMeshGeometry::jacobian

10.129.1.3 position

mfem::Vector serif::mapping::DMFixedMeshGeometry::position

10.129.1.4 weight

double serif::mapping::DMFixedMeshGeometry::weight {0.0}

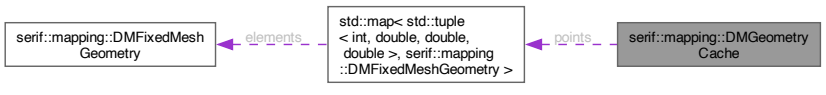
The documentation for this struct was generated from the following file:

- src/include/serif/mapping/data.hpp

10.130 serif::mapping::DMGeometryCache Struct Reference

#include <data.hpp>

Collaboration diagram for serif::mapping::DMGeometryCache:



Public Attributes

- bool enabled {true}
- const mfem::Mesh * mesh {nullptr}
- long sequence {-1}
- long nodesSequence {-1}
- std::map< std::tuple< int, double, double, double >, DMFixedMeshGeometry > points
- uint64_t hits {0}
- uint64_t misses {0}

10.130.1 Member Data Documentation

10.130.1.1 enabled

bool serif::mapping::DMGeometryCache::enabled {true}

10.130.1.2 hits

uint64_t serif::mapping::DMGeometryCache::hits {0}

10.130.1.3 mesh

const mfem::Mesh* serif::mapping::DMGeometryCache::mesh {nullptr}

10.130.1.4 misses

uint64_t serif::mapping::DMGeometryCache::misses {0}

10.130.1.5 nodesSequence

long serif::mapping::DMGeometryCache::nodesSequence {-1}

10.130.1.6 points

std::map<std::tuple<int, double, double, double>, DMFixedMeshGeometry> serif::mapping::DMGeometryCache::points

10.130.1.7 sequence

long serif::mapping::DMGeometryCache::sequence {-1}

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/data.hpp

10.131 serif::mapping::DMGeometryCacheStatistics Struct Reference

#include <data.hpp>

Public Attributes

- std::uint64_t hits {0}
- std::uint64_t misses {0}
- std::size_t points {0}

10.131.1 Member Data Documentation

10.131.1.1 hits

std::uint64_t serif::mapping::DMGeometryCacheStatistics::hits {0}

10.131.1.2 misses

std::uint64_t serif::mapping::DMGeometryCacheStatistics::misses {0}

10.131.1.3 points

std::size_t serif::mapping::DMGeometryCacheStatistics::points {0}

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/data.hpp

10.132 serif::mapping::DMReferenceDisplacementSample Struct Reference

#include <data.hpp>

Public Attributes

- const mfem::Vector & value
- const mfem::DenseMatrix & gradient

10.132.1 Member Data Documentation

10.132.1.1 gradient

const mfem::DenseMatrix& serif::mapping::DMReferenceDisplacementSample::gradient

10.132.1.2 value

const mfem::Vector& serif::mapping::DMReferenceDisplacementSample::value

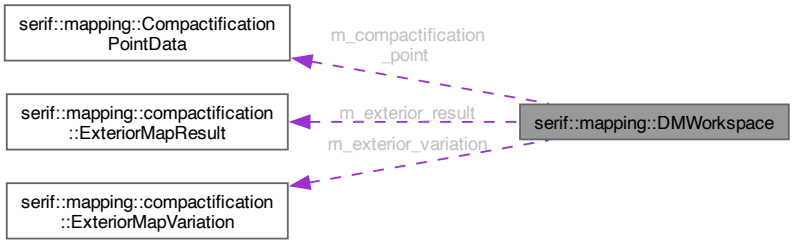
The documentation for this struct was generated from the following file:

- src/include/serif/mapping/data.hpp

10.133 serif::mapping::DMWorkspace Class Reference

#include <data.hpp>

Collaboration diagram for serif::mapping::DMWorkspace:



Public Member Functions

- DMWorkspace (int dimension=3)
- void set_dimension (int dimension)
- int get_dimension () const noexcept

Public Attributes

- mfem::Vector m_shape
- mfem::DenseMatrix m_reference_dshape
- mfem::DenseMatrix m_mesh_dshape
- mfem::DenseMatrix m_reference_field_jacobian
- mfem::Vector m_field_value
- mfem::DenseMatrix m_field_jacobian
- mfem::Vector m_compactification_shape
- mfem::DenseMatrix m_compactification_dshape
- CompactificationPointData m_compactification_point
- mfem::Vector m_reference_normal
- mfem::Vector m_mapped_normal
- mfem::DenseMatrix m_full_element_jacobian
- mfem::Vector m_vector_temp
- mfem::DenseMatrix m_matrix_temp_1
- mfem::DenseMatrix m_matrix_temp_2
- compactification::ExteriorMapResult m_exterior_result
- compactification::ExteriorMapVariation m_exterior_variation

Private Attributes

- int m_dimension

10.133.1 Constructor & Destructor Documentation

10.133.1.1 DMWorkspace()

serif::mapping::DMWorkspace::DMWorkspace (
int dimension = 3) [explicit]

10.133.2 Member Function Documentation

10.133.2.1 get_dimension()

int serif::mapping::DMWorkspace::get_dimension () const [nodiscard], [noexcept]

10.133.2.2 set_dimension()

void serif::mapping::DMWorkspace::set_dimension (
int dimension)

10.133.3 Member Data Documentation

10.133.3.1 m_compactification_dshape

mfem::DenseMatrix serif::mapping::DMWorkspace::m_compactification_dshape

10.133.3.2 m_compactification_point

CompactificationPointData serif::mapping::DMWorkspace::m_compactification_point

10.133.3.3 m_compactification_shape

mfem::Vector serif::mapping::DMWorkspace::m_compactification_shape

10.133.3.4 m_dimension

int serif::mapping::DMWorkspace::m_dimension [private]

10.133.3.5 m_exterior_result

compactification::ExteriorMapResult serif::mapping::DMWorkspace::m_exterior_result

10.133.3.6 m_exterior_variation

compactification::ExteriorMapVariation serif::mapping::DMWorkspace::m_exterior_variation

10.133.3.7 m_field_jacobian

mfem::DenseMatrix serif::mapping::DMWorkspace::m_field_jacobian

10.133.3.8 m_field_value

mfem::Vector serif::mapping::DMWorkspace::m_field_value

10.133.3.9 m_full_element_jacobian

mfem::DenseMatrix serif::mapping::DMWorkspace::m_full_element_jacobian

10.133.3.10 m_mapped_normal

mfem::Vector serif::mapping::DMWorkspace::m_mapped_normal

10.133.3.11 m_matrix_temp_1

mfem::DenseMatrix serif::mapping::DMWorkspace::m_matrix_temp_1

10.133.3.12 m_matrix_temp_2

mfem::DenseMatrix serif::mapping::DMWorkspace::m_matrix_temp_2

10.133.3.13 m_mesh_dshape

mfem::DenseMatrix serif::mapping::DMWorkspace::m_mesh_dshape

10.133.3.14 m_reference_dshape

mfem::DenseMatrix serif::mapping::DMWorkspace::m_reference_dshape

10.133.3.15 m_reference_field_jacobian

mfem::DenseMatrix serif::mapping::DMWorkspace::m_reference_field_jacobian

10.133.3.16 m_reference_normal

mfem::Vector serif::mapping::DMWorkspace::m_reference_normal

10.133.3.17 m_shape

mfem::Vector serif::mapping::DMWorkspace::m_shape

10.133.3.18 m_vector_temp

mfem::Vector serif::mapping::DMWorkspace::m_vector_temp

The documentation for this class was generated from the following files:

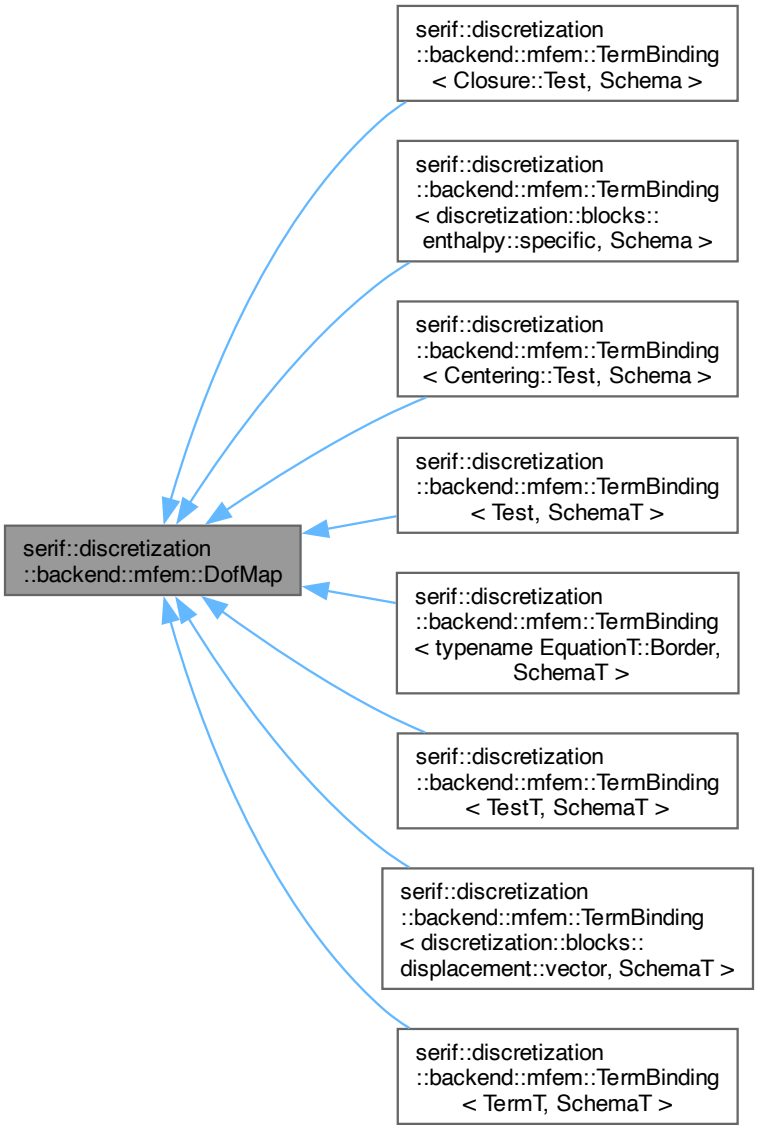
- src/include/serif/mapping/data.hpp
- src/lib/serif/mapping/data.cpp

10.134 serif::discretization::backend::mfem::DofMap Class Reference

A mapping from a full MFEM finite-element space to a reduced set of DOFs that are supporter by the given element support. That is to say that if one has some set of dofs D (perhaps it looks like [0, 1, 2, 3, 4] as a hypothetical) and dofs 0, 2, and 4 are associated to domain A while 1 and 3 are associated to domain B. Then if one were to construct a DofMap with element support [1, 0, 1, 0, 1] (where 1 indicates that the element is supported and 0 indicates that it is not) then the reduced dof map would be [0, 2, 4] and the full to reduced map would be [-1, -1, 0, -1, 1].

#include <dof_map.hpp>

Inheritance diagram for serif::discretization::backend::mfem::DofMap:



Public Member Functions

- DofMap (const DofMap &)=delete
- DofMap & operator= (const DofMap &)=delete
- DofMap (DofMap &&)=delete
- DofMap & operator= (DofMap &&)=delete
- ::mfem::FiniteElementSpace & space () const

  Return the borrowed finite-element space associated with this DOF map.

- ::mfem::Mesh & mesh () const noexcept

  Return the borrowed mesh associated with this DOF map.
- int full_size () const noexcept

  Return the number of full-space DOFs, including those not supported by the element support.
- int size () const noexcept

  Return the number of supported DOFs, which is the size of the reduced coefficient vector.
- std::optional< int > reduced_dof (const int full_dof) const

  Located a full space dof in the reduced space. E.g if I ask where is dof at index x in the full space this will tell me what index it is in the reduced space. If the dof is not supported this will return std::nullopt.
- bool supports (const int element) const

  Check whether an element is supported by this dof map.
- void validate () const

  Validate the dof map.
- void validate_values (const ::mfem::Vector &values) const

  Check supported coefficient count and finiteness.
- ::mfem::Vector gather (const ::mfem::Vector &full) const

  Restrict a full coefficient vector to supported DOFs.
- ::mfem::Vector scatter (const ::mfem::Vector &reduced) const

  Expand supported coefficients with zero outside their support.
- void element_values (const int element, const ::mfem::Vector &reduced, ::mfem::Vector &values) const

  Extract element coefficients, including primal orientation transforms.
- void add_element (const int element, ::mfem::Vector &values, ::mfem::Vector &full) const

  Apply dual orientation transforms and add an element residual.

Protected Member Functions

- DofMap (::mfem::FiniteElementSpace &space, const ::mfem::Array< int > &element_support)
- DofMap (::mfem::Mesh &mesh, const ::mfem::Array< int > &element_support)
- ~DofMap ()=default

Private Attributes

- ::mfem::FiniteElementSpace * m_space

  Borrowed pointer to the finite-element space associated with this dof map, or nullptr for a global scalar. This must outlive this class.
- ::mfem::Mesh & m_mesh

  Borrowed reference to the mesh associated with this dof map. This must outlive this class.
- long m_space_sequence

  Sequence number of the finite-element space at construction, or -1 for a global scalar.
- long m_mesh_sequence

  Sequence number of the mesh at construction.
- ::mfem::Array< int > m_attributes

  Array of attribute indices associated with this dof map.
- ::mfem::Array< int > m_element_support

  Array of 0/1 values indicating whether each element is supported by this dof map.
- ::mfem::Array< int > m_reduced_to_full

  Mapping from reduced to full degrees of freedom.
- ::mfem::Array< int > m_full_to_reduced

  Mapping from full to reduced degrees of freedom.

10.134.1 Detailed Description

A mapping from a full MFEM finite-element space to a reduced set of DOFs that are supporter by the given element support. That is to say that if one has some set of dofs D (perhaps it looks like [0, 1, 2, 3, 4] as a hypothetical) and dofs 0, 2, and 4 are associated to domain A while 1 and 3 are associated to domain B. Then if one were to construct a DofMap with element support [1, 0, 1, 0, 1] (where 1 indicates that the element is supported and 0 indicates that it is not) then the reduced dof map would be [0, 2, 4] and the full to reduced map would be [-1, -1, 0, -1, 1].

Note

This is a key place which will need to be updated if we are to support, as I believe we must eventually, non confirming meshes. Further, this current implementation does not support multi-rank parallelism in a smart manner.

10.134.2 Constructor & Destructor Documentation

10.134.2.1 DofMap() [1/4]

serif::discretization::backend::mfem::DofMap::DofMap (
::mfem::FiniteElementSpace & space,
const ::mfem::Array< int > & element_support) [inline], [protected]

10.134.2.2 DofMap() [2/4]

serif::discretization::backend::mfem::DofMap::DofMap (
::mfem::Mesh & mesh,
const ::mfem::Array< int > & element_support) [inline], [protected]

10.134.2.3 ~DofMap()

serif::discretization::backend::mfem::DofMap::~DofMap () [protected], [default]

10.134.2.4 DofMap() [3/4]

serif::discretization::backend::mfem::DofMap::DofMap (
const DofMap &) [delete]

10.134.2.5 DofMap() [4/4]

serif::discretization::backend::mfem::DofMap::DofMap (
DofMap &&) [delete]

10.134.3 Member Function Documentation

10.134.3.1 add_element()

void serif::discretization::backend::mfem::DofMap::add_element (
const int element,
::mfem::Vector & values,
::mfem::Vector & full) const [inline]

Apply dual orientation transforms and add an element residual.

Parameters

element	element index
values	vector of element coefficients
full	full coefficient vector to which the element residual will be added

10.134.3.2 element_values()

void serif::discretization::backend::mfem::DofMap::element_values (
const int element,
const ::mfem::Vector & reduced,
::mfem::Vector & values) const [inline]

Extract element coefficients, including primal orientation transforms.

Parameters

element	element index
reduced	vector of supported coefficients
values	vector to store extracted element coefficients

10.134.3.3 full_size()

int serif::discretization::backend::mfem::DofMap::full_size () const [inline], [nodiscard], [noexcept]

Return the number of full-space DOFs, including those not supported by the element support.

Returns

The number of full-space DOFs (not reduced by element support)

10.134.3.4 gather()

::mfem::Vector serif::discretization::backend::mfem::DofMap::gather (
const ::mfem::Vector & full) const [inline], [nodiscard]

Restrict a full coefficient vector to supported DOFs.

Parameters

full	full coefficient vector

Returns

vector of supported coefficients

10.134.3.5 mesh()

::mfem::Mesh & serif::discretization::backend::mfem::DofMap::mesh () const [inline], [nodiscard], [noexcept]

Return the borrowed mesh associated with this DOF map.

Returns

A reference to the mesh in m_mesh

10.134.3.6 operator=() [1/2]

DofMap & serif::discretization::backend::mfem::DofMap::operator= (
const DofMap &) [delete]

10.134.3.7 operator=() [2/2]

DofMap & serif::discretization::backend::mfem::DofMap::operator= (
DofMap &&) [delete]

10.134.3.8 reduced_dof()

std::optional< int > serif::discretization::backend::mfem::DofMap::reduced_dof (
const int full_dof) const [inline], [nodiscard]

Located a full space dof in the reduced space. E.g if I ask where is dof at index x in the full space this will tell me what index it is in the reduced space. If the dof is not supported this will return std::nullopt.

Parameters

full_dof	index of the dof in the full space

Returns

An optional index of the dof in the reduced space, or std::nullopt if the dof is not supported.

10.134.3.9 scatter()

::mfem::Vector serif::discretization::backend::mfem::DofMap::scatter (
const ::mfem::Vector & reduced) const [inline], [nodiscard]

Expand supported coefficients with zero outside their support.

Parameters

reduced	vector of supported coefficients

Returns

full coefficient vector with zeros outside the support

10.134.3.10 size()

int serif::discretization::backend::mfem::DofMap::size () const [inline], [nodiscard], [noexcept]

Return the number of supported DOFs, which is the size of the reduced coefficient vector.

Returns

The number of supported DOFs (reduced by element support)

10.134.3.11 space()

::mfem::FiniteElementSpace & serif::discretization::backend::mfem::DofMap::space () const [inline], [nodiscard]

Return the borrowed finite-element space associated with this DOF map.

Returns

A reference to the finite-element space in m_space

10.134.3.12 supports()

bool serif::discretization::backend::mfem::DofMap::supports (
const int element) const [inline], [nodiscard]

Check whether an element is supported by this dof map.

Parameters

element	element index

Returns

true if element is supported by this dof map, false otherwise

10.134.3.13 validate()

void serif::discretization::backend::mfem::DofMap::validate () const [inline]

Validate the dof map.

10.134.3.14 validate_values()

void serif::discretization::backend::mfem::DofMap::validate_values (
const ::mfem::Vector & values) const [inline]

Check supported coefficient count and finiteness.

Parameters

values	vector of coefficients to validate

10.134.4 Member Data Documentation

10.134.4.1 m_attributes

::mfem::Array<int> serif::discretization::backend::mfem::DofMap::m_attributes [private]

Array of attribute indices associated with this dof map.

10.134.4.2 m_element_support

::mfem::Array<int> serif::discretization::backend::mfem::DofMap::m_element_support [private]

Array of 0/1 values indicating whether each element is supported by this dof map.

10.134.4.3 m_full_to_reduced

::mfem::Array<int> serif::discretization::backend::mfem::DofMap::m_full_to_reduced [private]

Mapping from full to reduced degrees of freedom.

10.134.4.4 m_mesh

::mfem::Mesh& serif::discretization::backend::mfem::DofMap::m_mesh [private]

Borrowed reference to the mesh associated with this dof map. This must outlive this class.

10.134.4.5 m_mesh_sequence

long serif::discretization::backend::mfem::DofMap::m_mesh_sequence [private]

Sequence number of the mesh at construction.

10.134.4.6 m_reduced_to_full

::mfem::Array<int> serif::discretization::backend::mfem::DofMap::m_reduced_to_full [private]

Mapping from reduced to full degrees of freedom.

10.134.4.7 m_space

::mfem::FiniteElementSpace* serif::discretization::backend::mfem::DofMap::m_space [private]

Borrowed pointer to the finite-element space associated with this dof map, or nullptr for a global scalar. This must outlive this class.

10.134.4.8 m_space_sequence

long serif::discretization::backend::mfem::DofMap::m_space_sequence [private]

Sequence number of the finite-element space at construction, or -1 for a global scalar.

The documentation for this class was generated from the following file:

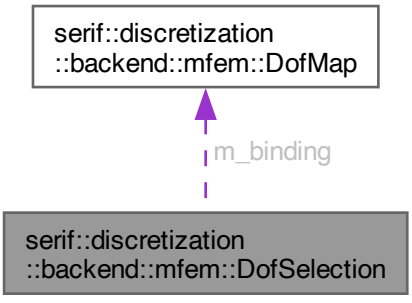
- src/include/serif/discretization/backend/mfem/dof_map.hpp

10.135 serif::discretization::backend::mfem::DofSelection Class Reference

A selection of supported DOFs from a term binding, with boundary attribute information. The purpose here is to allow for the selection of a subset of DOFs from a term binding, such as those associated with a particular boundary, while still retaining the ability to validate the selection against the underlying mesh and term binding.

#include <dof_selection.hpp>

Collaboration diagram for serif::discretization::backend::mfem::DofSelection:



Public Member Functions

- DofSelection (const DofMap &binding, const ::mfem::Array< int > &reduced_dofs, const bool track↩ _coordinates=false)

  Construct a DOF selection from a term binding and a list of reduced DOFs.
- const DofMap & binding () const noexcept

  Return the underlying term binding associated with this DOF selection.
- const ::mfem::Array< int > & reduced_dofs () const noexcept

  Return the list of reduced DOFs in this selection.
- int size () const noexcept

  Return the number of selected DOFs in this selection.
- const ::mfem::Array< int > & boundary_attributes () const noexcept

  Return the list of boundary attributes associated with this selection.
- bool tracks_coordinates () const noexcept

  Report whether this selection becomes stale when the fixed mesh coordinates change. Point constraints require this tracking even when a caller supplies the selected DOFs explicitly.
- void validate () const

  Validate the DOF selection against the underlying term binding and mesh. This checks that the underlying term binding and mesh have not changed since the DOF selection was constructed. If they have changed, an exception is thrown. This is important to ensure that the DOF selection remains valid and consistent with the underlying data structures.

Private Attributes

- const DofMap & m_binding

  Reference to the underlying term binding. This must outlive the DOF selection.
- ::mfem::Array< int > m_dofs

  List of reduced DOFs in this selection. These are in the reduced space of the binding provided at construction time.
- ::mfem::Array< int > m_boundary_attributes

  List of boundary attributes associated with this selection. These are in the same order as the boundaries in the underlying mesh.
- long m_nodes_sequence

  Coordinate revision, or -1 for selections independent of geometry.

10.135.1 Detailed Description

A selection of supported DOFs from a term binding, with boundary attribute information. The purpose here is to allow for the selection of a subset of DOFs from a term binding, such as those associated with a particular boundary, while still retaining the ability to validate the selection against the underlying mesh and term binding.

10.135.2 Constructor & Destructor Documentation

10.135.2.1 DofSelection()

```
serif::discretization::backend::mfem::DofSelection::DofSelection (
        const DofMap & binding,
        const ::mfem::Array< int > & reduced_dofs,
        const bool track_coordinates = false)   [inline]
```

Construct a DOF selection from a term binding and a list of reduced DOFs.

Parameters

binding	A reference to the term binding from which the DOFs are selected. Further, this binding must outlive the DOF selection.
reduced_dofs	A list of reduced DOFs to select. This list should be in the reduced space of the term binding. Further, these must be sorted in a strictly increasing order and must be unique. This is enforced at construction time.

10.135.3 Member Function Documentation

10.135.3.1 binding()

const DofMap & serif::discretization::backend::mfem::DofSelection::binding () const [inline], [nodiscard], [noexcept]

Return the underlying term binding associated with this DOF selection.

Returns

A reference to the underlying term binding.

10.135.3.2 boundary_attributes()

const ::mfem::Array< int > & serif::discretization::backend::mfem::DofSelection::boundary_attributes () const [inline], [nodiscard], [noexcept]

Return the list of boundary attributes associated with this selection.

Returns

A reference to the list of boundary attributes. These are in the same order as the boundaries in the underlying mesh.

10.135.3.3 reduced_dofs()

const ::mfem::Array< int > & serif::discretization::backend::mfem::DofSelection::reduced_dofs () const [inline], [nodiscard], [noexcept]

Return the list of reduced DOFs in this selection.

Returns

A reference to the list of reduced DOFs. Note these are in the reduced space of the binding provided at construction time.

10.135.3.4 size()

int serif::discretization::backend::mfem::DofSelection::size () const [inline], [nodiscard], [noexcept]

Return the number of selected DOFs in this selection.

Returns

The number of selected DOFs.

10.135.3.5 tracks_coordinates()

bool serif::discretization::backend::mfem::DofSelection::tracks_coordinates () const [inline], [nodiscard], [noexcept]

Report whether this selection becomes stale when the fixed mesh coordinates change. Point constraints require this tracking even when a caller supplies the selected DOFs explicitly.

Returns

True if coordinate updates invalidate the selection.

10.135.3.6 validate()

void serif::discretization::backend::mfem::DofSelection::validate () const [inline]

Validate the DOF selection against the underlying term binding and mesh. This checks that the underlying term binding and mesh have not changed since the DOF selection was constructed. If they have changed, an exception is thrown. This is important to ensure that the DOF selection remains valid and consistent with the underlying data structures.

10.135.4 Member Data Documentation

10.135.4.1 m_binding

const DofMap& serif::discretization::backend::mfem::DofSelection::m_binding [private]

Reference to the underlying term binding. This must outlive the DOF selection.

10.135.4.2 m_boundary_attributes

::mfem::Array<int> serif::discretization::backend::mfem::DofSelection::m_boundary_attributes [private]

List of boundary attributes associated with this selection. These are in the same order as the boundaries in the underlying mesh.

10.135.4.3 m_dofs

::mfem::Array<int> serif::discretization::backend::mfem::DofSelection::m_dofs [private]

List of reduced DOFs in this selection. These are in the reduced space of the binding provided at construction time.

10.135.4.4 m_nodes_sequence

long serif::discretization::backend::mfem::DofSelection::m_nodes_sequence [private]

Coordinate revision, or -1 for selections independent of geometry.

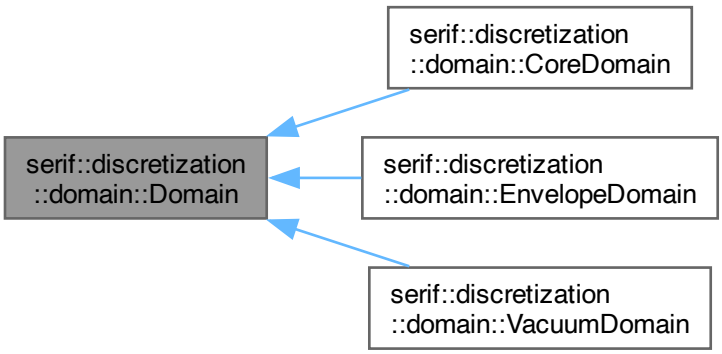
The documentation for this class was generated from the following file:

- src/include/serif/discretization/backend/mfem/dof_selection.hpp

10.136 serif::discretization::domain::Domain Struct Reference

#include <types.hpp>

Inheritance diagram for serif::discretization::domain::Domain:



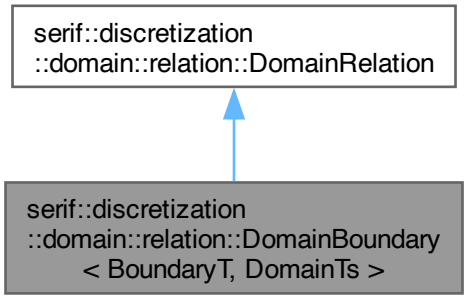
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/types.hpp

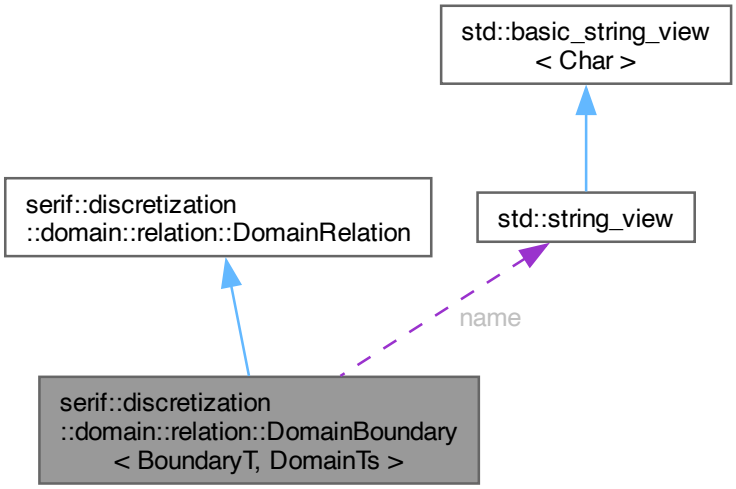
10.137 serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs > Struct Template Reference

#include <relations.hpp>

Inheritance diagram for serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >:



Collaboration diagram for serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >:



Public Types

- using boundary_type = BoundaryT
- using domain_types = DomainOrSetList<DomainTs...>

Static Public Attributes

- static constexpr std::size_t domain_count = CountVariadicArguments<DomainTs...>()
- static constexpr std::string_view name = "domain_boundary"

10.137.1 Member Typedef Documentation

10.137.1.1 boundary_type

template<IsBoundary BoundaryT, IsDomainOrSet... DomainTs>
using serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >::boundary_type = BoundaryT

10.137.1.2 domain_types

template<IsBoundary BoundaryT, IsDomainOrSet... DomainTs>
using serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >::domain_types = DomainOrSetList<Domain↩
Ts...>

10.137.2 Member Data Documentation

10.137.2.1 domain_count

template<IsBoundary BoundaryT, IsDomainOrSet... DomainTs>
std::size_t serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >::domain_count = CountVariadicArguments<Domain
Ts...>() [static], [constexpr]

10.137.2.2 name

template<IsBoundary BoundaryT, IsDomainOrSet... DomainTs>
std::string_view serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >::name = "domain_↩
boundary" [static], [constexpr]

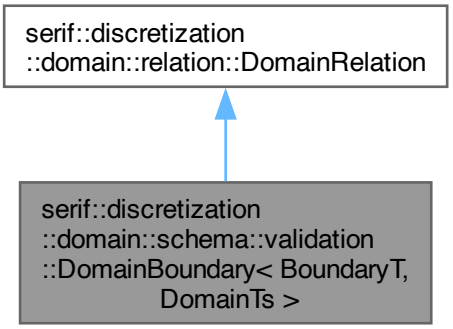
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/relations.hpp

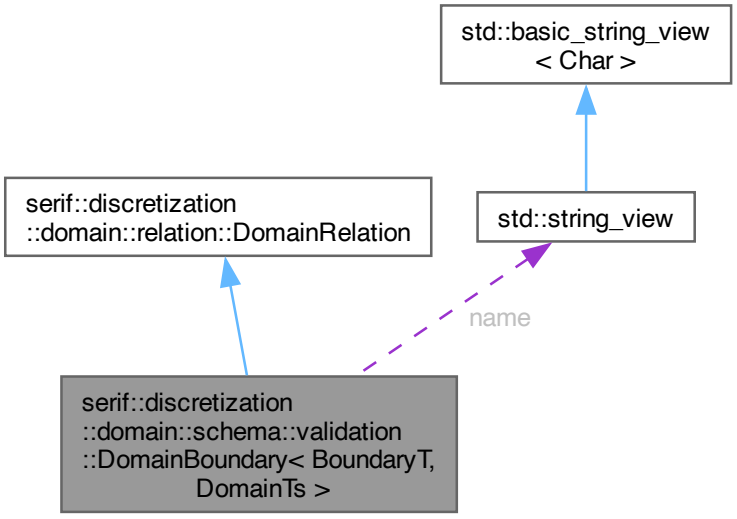
10.138 serif::discretization::domain::schema::validation::DomainBoundary< BoundaryT, DomainTs > Struct Template Reference

#include <relations.hpp>

Inheritance diagram for serif::discretization::domain::schema::validation::DomainBoundary< BoundaryT, DomainTs >:



Collaboration diagram for serif::discretization::domain::schema::validation::DomainBoundary< BoundaryT, DomainTs >:



Public Types

- using boundary_type = BoundaryT
- using domain_types = DomainOrSetList<DomainTs...>

Static Public Attributes

- static constexpr std::size_t domain_count = CountVariadicArguments<DomainTs...>()
- static constexpr std::string_view name = "domain_boundary"

10.138.1 Member Typedef Documentation

10.138.1.1 boundary_type

template<IsBoundary BoundaryT, IsDomainOrSet... DomainTs>
using serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >::boundary_type = BoundaryT

10.138.1.2 domain_types

template<IsBoundary BoundaryT, IsDomainOrSet... DomainTs>
using serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >::domain_types = DomainOrSetList<Domain↩
Ts...>

10.138.2 Member Data Documentation

10.138.2.1 domain_count

template<IsBoundary BoundaryT, IsDomainOrSet... DomainTs>
std::size_t serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >::domain_count = CountVariadicArguments<Domain
Ts...>() [static], [constexpr]

10.138.2.2 name

template<IsBoundary BoundaryT, IsDomainOrSet... DomainTs>
std::string_view serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >::name = "domain_↩
boundary" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/relations.hpp

10.139 serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics Struct Reference

#include <runtime.hpp>

Public Attributes

- int faceID {-1}
- int boundaryElementID {-1}
- int expectedBoundaryID {0}
- std::optional< int > actualBoundaryID = std::nullopt
- int firstElementID {-1}
- int secondElementID {-1}
- std::optional< int > firstDomainID = std::nullopt
- std::optional< int > secondDomainID = std::nullopt

10.139.1 Member Data Documentation

10.139.1.1 actualBoundaryID

std::optional<int> serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::actualBoundaryID = std::nullopt

10.139.1.2 boundaryElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::boundaryElementID {-1}

10.139.1.3 expectedBoundaryID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::expectedBoundaryID {0}

10.139.1.4 faceID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::faceID {-1}

10.139.1.5 firstDomainID

std::optional<int> serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::firstDomainID = std::nullopt

10.139.1.6 firstElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::firstElementID {-1}

10.139.1.7 secondDomainID

std::optional<int> serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::secondDomainID = std::nullopt

10.139.1.8 secondElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::secondElementID {-1}

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/runtime.hpp

10.140 serif::discretization::domain::schema::validation::RelationValidationResult::DomainBoundaryDiagnostics Struct Reference

#include <runtime.hpp>

Public Attributes

- int faceID {-1}
- int boundaryElementID {-1}
- int expectedBoundaryID {0}
- std::optional< int > actualBoundaryID = std::nullopt
- int firstElementID {-1}
- int secondElementID {-1}
- std::optional< int > firstDomainID = std::nullopt
- std::optional< int > secondDomainID = std::nullopt

10.140.1 Member Data Documentation

10.140.1.1 actualBoundaryID

std::optional<int> serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::actualBoundaryID = std::nullopt

10.140.1.2 boundaryElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::boundary↩ ElementID {-1}

10.140.1.3 expectedBoundaryID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::expected↩ BoundaryID {0}

10.140.1.4 faceID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::faceID {-1}

10.140.1.5 firstDomainID

std::optional<int> serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics↩ ::firstDomainID = std::nullopt

10.140.1.6 firstElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::firstElementID {-1}

10.140.1.7 secondDomainID

std::optional<int> serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics↩ ::secondDomainID = std::nullopt

10.140.1.8 secondElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics::secondElementID {-1}

The documentation for this struct was generated from the following file:

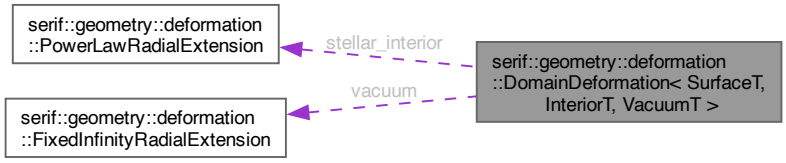
- src/include/serif/discretization/domain/relation/validation/runtime.hpp

10.141 serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT > Struct Template Reference

State how the stellar surface moves and how that motion extends into the two surrounding regions.

#include <models.hpp>

Collaboration diagram for serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >:



Public Member Functions

- bool operator== (const DomainDeformation &) const =default
  Compare physical choices without attaching preparation state to the model.

Public Attributes

- SurfaceT surface
- InteriorT stellar_interior
- VacuumT vacuum

10.141.1 Detailed Description

template<typename SurfaceT, typename InteriorT = PowerLawRadialExtension, typename VacuumT = FixedInfinityRadialExtension>
struct serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >

State how the stellar surface moves and how that motion extends into the two surrounding regions.

10.141.2 Member Function Documentation

10.141.2.1 operator==()

template<typename SurfaceT, typename InteriorT = PowerLawRadialExtension, typename VacuumT = FixedInfinityRadialExtension>
bool serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >::operator== (
const DomainDeformation< SurfaceT, InteriorT, VacuumT > &) const [nodiscard], [default]

Compare physical choices without attaching preparation state to the model.

10.141.3 Member Data Documentation

10.141.3.1 stellar_interior

template<typename SurfaceT, typename InteriorT = PowerLawRadialExtension, typename VacuumT = FixedInfinityRadial↩
Extension>
InteriorT serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >::stellar_interior

10.141.3.2 surface

template<typename SurfaceT, typename InteriorT = PowerLawRadialExtension, typename VacuumT = FixedInfinityRadial↩
Extension>
SurfaceT serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >::surface

10.141.3.3 vacuum

template<typename SurfaceT, typename InteriorT = PowerLawRadialExtension, typename VacuumT = FixedInfinityRadial↩
Extension>
VacuumT serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >::vacuum

The documentation for this struct was generated from the following file:

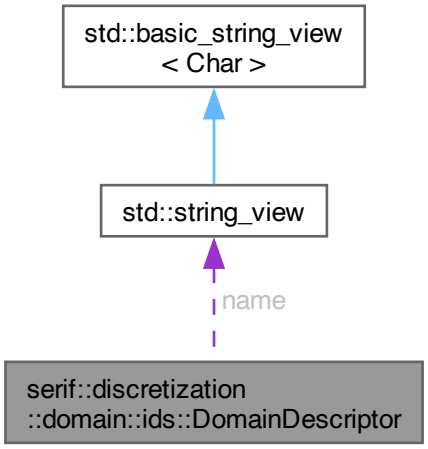
- src/include/serif/geometry/deformation/models.hpp

10.142 serif::discretization::domain::ids::DomainDescriptor Struct Reference

The DomainDescriptor struct is a runtime representation of a domain, containing its name and integer attribute ID. This is used to project compile-time DomainID types into runtime values.

#include <domain.hpp>

Collaboration diagram for serif::discretization::domain::ids::DomainDescriptor:



Public Attributes

- std::string_view name

  The name of the domain, used for runtime identification.
- int ID

  The integer attribute value in the mesh file that corresponds to this domain.

10.142.1 Detailed Description

The DomainDescriptor struct is a runtime representation of a domain, containing its name and integer attribute ID. This is used to project compile-time DomainID types into runtime values.

10.142.2 Member Data Documentation

10.142.2.1 ID

int serif::discretization::domain::ids::DomainDescriptor::ID

The integer attribute value in the mesh file that corresponds to this domain.

10.142.2.2 name

std::string_view serif::discretization::domain::ids::DomainDescriptor::name

The name of the domain, used for runtime identification.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/domain.hpp

10.143 serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue > Struct Template Reference

The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation.

#include <domain.hpp>

Public Types

- using domain_type = DomainT
  The semantic domain type associated with this DomainID.

Static Public Attributes

- static constexpr int ID = IdentifierValue
  The integer attribute value in the mesh file that corresponds to this domain.

10.143.1 Detailed Description

template<IsDomain DomainT, int IdentifierValue>
struct serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >

The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation.

Template Parameters

DomainT	The semantic domain type (must satisfy the IsDomain concept).
IdentifierValue	The integer attribute value in the mesh file that corresponds to this domain.

10.143.2 Member Typedef Documentation

10.143.2.1 domain_type

template<IsDomain DomainT, int IdentifierValue>
using serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >::domain_type = DomainT

The semantic domain type associated with this DomainID.

10.143.3 Member Data Documentation

10.143.3.1 ID

template<IsDomain DomainT, int IdentifierValue>
int serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >::ID = IdentifierValue [static], [constexpr]

The integer attribute value in the mesh file that corresponds to this domain.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/domain.hpp

10.144 serif::discretization::domain::schema::DomainID< DomainT, IdentifierValue > Struct Template Reference

The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation.

#include <domain.hpp>

Public Types

- using domain_type = DomainT
  The semantic domain type associated with this DomainID.

Static Public Attributes

- static constexpr int ID = IdentifierValue
  The integer attribute value in the mesh file that corresponds to this domain.

10.144.1 Detailed Description

template<IsDomain DomainT, int IdentifierValue>
struct serif::discretization::domain::schema::DomainID< DomainT, IdentifierValue >

The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation.

Template Parameters

DomainT	The semantic domain type (must satisfy the IsDomain concept).
IdentifierValue	The integer attribute value in the mesh file that corresponds to this domain.

10.144.2 Member Typedef Documentation

10.144.2.1 domain_type

template<IsDomain DomainT, int IdentifierValue>
using serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >::domain_type = DomainT

The semantic domain type associated with this DomainID.

10.144.3 Member Data Documentation

10.144.3.1 ID

template<IsDomain DomainT, int IdentifierValue>
int serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >::ID = IdentifierValue [static], [constexpr]

The integer attribute value in the mesh file that corresponds to this domain.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/domain.hpp

10.145 serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs > Struct Template Reference

A list of domain IDs in the discretization framework.

#include <lists.hpp>

Static Public Member Functions

- static constexpr std::array< DomainDescriptor, count > descriptors () noexcept
  Get an array of DomainDescriptor objects for the domain IDs in the list.

Static Public Attributes

- static constexpr std::size_t count = CountVariadicArguments<DomainIDTs...>()
  The number of domain IDs in the list.

10.145.1 Detailed Description

template<IsDomainID... DomainIDTs>
requires validation::AreDistinctDomainIDs<DomainIDTs...>
struct serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >

A list of domain IDs in the discretization framework.

10.145.2 Member Function Documentation

10.145.2.1 descriptors()

template<IsDomainID... DomainIDTs>
static constexpr std::array< DomainDescriptor, count > serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >::descriptors () [inline], [static], [nodiscard], [constexpr], [noexcept]

Get an array of DomainDescriptor objects for the domain IDs in the list.

Returns

An array of DomainDescriptor objects, each containing the name and ID of a domain.

10.145.3 Member Data Documentation

10.145.3.1 count

template<IsDomainID... DomainIDTs>
std::size_t serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >::count = CountVariadicArguments<DomainIDTs...>() [static], [constexpr]

The number of domain IDs in the list.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/lists/lists.hpp

10.146 serif::discretization::domain::schema::DomainIDList< DomainIDTs > Struct Template Reference

A list of domain IDs in the discretization framework.

#include <lists.hpp>

Static Public Member Functions

- static constexpr std::array< DomainDescriptor, count > descriptors () noexcept
  Get an array of DomainDescriptor objects for the domain IDs in the list.

Static Public Attributes

- static constexpr std::size_t count = CountVariadicArguments<DomainIDTs...>()
  The number of domain IDs in the list.

10.146.1 Detailed Description

template<IsDomainID... DomainIDTs>
requires validation::AreDistinctDomainIDs<DomainIDTs...>
struct serif::discretization::domain::schema::DomainIDList< DomainIDTs >

A list of domain IDs in the discretization framework.

10.146.2 Member Function Documentation

10.146.2.1 descriptors()

template<IsDomainID... DomainIDTs>
static constexpr std::array< DomainDescriptor, count > serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >::descriptors () [inline], [static], [nodiscard], [constexpr], [noexcept]

Get an array of DomainDescriptor objects for the domain IDs in the list.

Returns

An array of DomainDescriptor objects, each containing the name and ID of a domain.

10.146.3 Member Data Documentation

10.146.3.1 count

template<IsDomainID... DomainIDTs>
std::size_t serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >::count = CountVariadicArguments<Domain↩IDTs...>() [static], [constexpr]

The number of domain IDs in the list.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/ids/lists/lists.hpp

10.147 serif::discretization::domain::DomainIDResolver< DomainT, DomainIDListT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/resolver.hpp

10.148 serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... > > Struct Template Reference

#include <resolver.hpp>

Public Types

- template<typename D>
  using ResolverFor = DomainIDResolver<D, ids::lists::DomainIDList<DomainIDTs...>>

Static Public Member Functions

- static constexpr bool contains_id (const int domainID) noexcept

Static Public Attributes

- static constexpr bool registered = (ResolverFor<DomainTs>::registered && ...)

10.148.1 Member Typedef Documentation

10.148.1.1 ResolverFor

template<IsDomain... DomainTs, ids::IsDomainID... DomainIDTs>
template<typename D>
using serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... > >::ResolverFor = DomainIDResolver<D, ids::lists::DomainIDList<DomainIDTs...>>

10.148.2 Member Function Documentation

10.148.2.1 contains_id()

template<IsDomain... DomainTs, ids::IsDomainID... DomainIDTs>
static constexpr bool serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... > >::contains_id (
const int domainID) [inline], [static], [nodiscard], [constexpr], [noexcept]

10.148.3 Member Data Documentation

10.148.3.1 registered

template<IsDomain... DomainTs, ids::IsDomainID... DomainIDTs>
bool serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... > >::registered = (ResolverFor<DomainTs>::registered && ...) [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/resolver.hpp

10.149 serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > > Struct Template Reference

#include <resolver.hpp>

Static Public Member Functions

- static constexpr bool contains_id (const int domainID) noexcept
- static consteval int id ()

Static Public Attributes

- static constexpr bool registered = (DomainIsIdentifiedBy<DomainT, DomainIDTs> || ...)

10.149.1 Member Function Documentation

10.149.1.1 contains_id()

template<IsDomain DomainT, ids::IsDomainID... DomainIDTs>
static constexpr bool serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > >::contains_id (
const int domainID) [inline], [static], [nodiscard], [constexpr], [noexcept]

10.149.1.2 id()

template<IsDomain DomainT, ids::IsDomainID... DomainIDTs>
static consteval int serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > >::id () [inline], [static], [nodiscard], [consteval]

10.149.2 Member Data Documentation

10.149.2.1 registered

template<IsDomain DomainT, ids::IsDomainID... DomainIDTs>
bool serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > >::registered = (DomainIsIdentifiedBy<DomainT, DomainIDTs> || ...) [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/resolver.hpp

10.150 serif::geometry::deformation::keys::DomainMap Struct Reference

#include <computations.hpp>

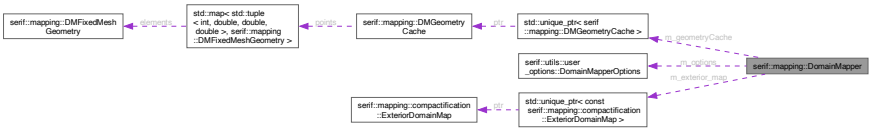
The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

10.151 serif::mapping::DomainMapper Class Reference

#include <domain_mapper.hpp>

Collaboration diagram for serif::mapping::DomainMapper:



Public Member Functions

- DomainMapper (const DomainMapper &)=delete
- DomainMapper & operator= (const DomainMapper &)=delete
- ~DomainMapper ()=default
- DomainMapper (DomainMapper &&)=default
- DomainMapper & operator= (DomainMapper &&)=default
- DomainMapper (utils::user_options::DomainMapperOptions options, std::unique_ptr< const compactification::ExteriorDomainMap > exterior_map)
- std::unique_ptr< DomainMapper > clone () const
- void set_geometry_cache_enabled (bool enabled) const
- bool is_geometry_cache_enabled () const noexcept

- void clear_geometry_cache () const
- DMGeometryCacheStatistics get_geometry_cache_statistics () const
- const DMFixedMeshGeometry * get_fixed_mesh_geometry (mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip) const
- MappingStatus evaluate_point (const ElementMappingData &element_data, mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip, DMWorkspace &workspace, MappingPointContext &context) const
- MappingStatus evaluate_volume (const ElementMappingData &element_data, mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip, DMWorkspace &workspace, VolumeMappingContext &context, const DMReferenceDisplacementSample *displacement_sample=nullptr) const
- MappingStatus evaluate_face (const ElementMappingData &element_data, mfem::FaceElementTransformations &transformation, FaceElementSide side, const mfem::IntegrationPoint &ip, DMWorkspace &workspace, FaceMappingContext &context) const
- MappingStatus evaluate_point_variation (const ElementMappingData &element_data, const ElementDisplacementData &direction, mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip, const MappingPointContext &context, DMWorkspace &workspace, MappingPointVariation &variation) const
- MappingStatus evaluate_volume_variation (const ElementMappingData &element_data, const ElementDisplacementData &direction, mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip, const VolumeMappingContext &context, DMWorkspace &workspace, VolumeMappingVariation &variation, const DMReferenceDisplacementSample *direction_sample=nullptr) const
- MappingStatus evaluate_face_variation (const ElementMappingData &element_data, const ElementDisplacementData &direction, mfem::FaceElementTransformations &transformation, FaceElementSide side, const mfem::IntegrationPoint &ip, const FaceMappingContext &context, DMWorkspace &workspace, FaceMappingVariation &variation) const
- bool is_compactified_element (const mfem::ElementTransformation &transformation) const noexcept
- int get_dimension () const noexcept
- const compactification::ExteriorDomainMap & get_exterior_map () const noexcept

Private Member Functions

- MappingStatus evaluate_point_impl (const ElementMappingData &element_data, mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip, DMWorkspace &workspace, MappingPointContext &context, const DMFixedMeshGeometry *geometry, const DMReferenceDisplacementSample *displacement_sample=nullptr) const
- void validate_element_data (const ElementMappingData &element_data) const
- void evaluate_displacement_field (const ElementDisplacementData &field, mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip, DMWorkspace &workspace, mfem::Vector &value, mfem::DenseMatrix &jacobian, const mfem::DenseMatrix *inverse_mesh_jacobian) const
- MappingStatus evaluate_compactification_coordinate (const ElementCompactificationData &compactification, mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip, DMWorkspace &workspace, CompactificationPointData &point_data, const mfem::DenseMatrix *inverse_mesh_jacobian) const
- MappingStatus evaluate_point_variation_impl (const ElementMappingData &element_data, const ElementDisplacementData &direction, mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip, const MappingPointContext &context, DMWorkspace &workspace, MappingPointVariation &variation, const mfem::DenseMatrix *inverse_mesh_jacobian, const DMReferenceDisplacementSample *direction_sample=nullptr) const

Static Private Member Functions

- static mfem::ElementTransformation & select_face_element_transformation (const mfem::Face↩ ElementTransformations &transformations, FaceElementSide side)
- static const mfem::IntegrationPoint & select_face_element_integration_point (const mfem::Face↩ ElementTransformations &transformations, FaceElementSide side)

Private Attributes

- std::unique_ptr< DMGeometryCache > m_geometryCache
- utils::user_options::DomainMapperOptions m_options
- std::unique_ptr< const compactification::ExteriorDomainMap > m_exterior_map

10.151.1 Constructor & Destructor Documentation

10.151.1.1 DomainMapper() [1/3]

serif::mapping::DomainMapper::DomainMapper (
const DomainMapper &) [delete]

10.151.1.2 ~DomainMapper()

serif::mapping::DomainMapper::~DomainMapper () [default]

10.151.1.3 DomainMapper() [2/3]

serif::mapping::DomainMapper::DomainMapper (
DomainMapper &&) [default]

10.151.1.4 DomainMapper() [3/3]

serif::mapping::DomainMapper::DomainMapper (
utils::user_options::DomainMapperOptions options,
std::unique_ptr< const compactification::ExteriorDomainMap > exterior_map)

10.151.2 Member Function Documentation

10.151.2.1 clear_geometry_cache()

void serif::mapping::DomainMapper::clear_geometry_cache () const

10.151.2.2 clone()

std::unique_ptr< DomainMapper > serif::mapping::DomainMapper::clone () const

10.151.2.3 evaluate_compactification_coordinate()

MappingStatus serif::mapping::DomainMapper::evaluate_compactification_coordinate (
const ElementCompactificationData & compactification,
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip,
DMWorkspace & workspace,
CompactificationPointData & point_data,
const mfem::DenseMatrix * inverse_mesh_jacobian) const [nodiscard], [private]

10.151.2.4 evaluate_displacement_field()

void serif::mapping::DomainMapper::evaluate_displacement_field (
const ElementDisplacementData & field,
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip,
DMWorkspace & workspace,
mfem::Vector & value,
mfem::DenseMatrix & jacobian,
const mfem::DenseMatrix * inverse_mesh_jacobian) const [private]

10.151.2.5 evaluate_face()

MappingStatus serif::mapping::DomainMapper::evaluate_face (
const ElementMappingData & element_data,
mfem::FaceElementTransformations & transformation,
FaceElementSide side,
const mfem::IntegrationPoint & ip,
DMWorkspace & workspace,
FaceMappingContext & context) const [nodiscard]

10.151.2.6 evaluate_face_variation()

MappingStatus serif::mapping::DomainMapper::evaluate_face_variation (
const ElementMappingData & element_data,
const ElementDisplacementData & direction,
mfem::FaceElementTransformations & transformation,
FaceElementSide side,
const mfem::IntegrationPoint & ip,
const FaceMappingContext & context,
DMWorkspace & workspace,
FaceMappingVariation & variation) const [nodiscard]

10.151.2.7 evaluate_point()

MappingStatus serif::mapping::DomainMapper::evaluate_point (
const ElementMappingData & element_data,
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip,
DMWorkspace & workspace,
MappingPointContext & context) const [nodiscard]

10.151.2.8 evaluate_point_impl()

MappingStatus serif::mapping::DomainMapper::evaluate_point_impl (
const ElementMappingData & element_data,
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip,
DMWorkspace & workspace,
MappingPointContext & context,
const DMFixedMeshGeometry * geometry,
const DMReferenceDisplacementSample * displacement_sample = nullptr) const [nodiscard], [private]

10.151.2.9 evaluate_point_variation()

MappingStatus serif::mapping::DomainMapper::evaluate_point_variation (
const ElementMappingData & element_data,
const ElementDisplacementData & direction,
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip,
const MappingPointContext & context,
DMWorkspace & workspace,
MappingPointVariation & variation) const [nodiscard]

10.151.2.10 evaluate_point_variation_impl()

MappingStatus serif::mapping::DomainMapper::evaluate_point_variation_impl (
const ElementMappingData & element_data,
const ElementDisplacementData & direction,
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip,
const MappingPointContext & context,
DMWorkspace & workspace,
MappingPointVariation & variation,
const mfem::DenseMatrix * inverse_mesh_jacobian,
const DMReferenceDisplacementSample * direction_sample = nullptr) const [nodiscard], [private]

10.151.2.11 evaluate_volume()

MappingStatus serif::mapping::DomainMapper::evaluate_volume (
const ElementMappingData & element_data,
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip,
DMWorkspace & workspace,
VolumeMappingContext & context,
const DMReferenceDisplacementSample * displacement_sample = nullptr) const [nodiscard]

10.151.2.12 evaluate_volume_variation()

MappingStatus serif::mapping::DomainMapper::evaluate_volume_variation (
const ElementMappingData & element_data,
const ElementDisplacementData & direction,
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip,
const VolumeMappingContext & context,
DMWorkspace & workspace,
VolumeMappingVariation & variation,
const DMReferenceDisplacementSample * direction_sample = nullptr) const [nodiscard]

Differentiate the physical volume map about the supplied prepared context. Optional direction samples contain the displacement value and its canonical-element gradient.

10.151.2.13 get_dimension()

int serif::mapping::DomainMapper::get_dimension () const [nodiscard], [noexcept]

10.151.2.14 get_exterior_map()

const compactification::ExteriorDomainMap & serif::mapping::DomainMapper::get_exterior_map () const [nodiscard], [noexcept]

10.151.2.15 get_fixed_mesh_geometry()

const DMFixedMeshGeometry * serif::mapping::DomainMapper::get_fixed_mesh_geometry (
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip) const [nodiscard]

10.151.2.16 get_geometry_cache_statistics()

DMGeometryCacheStatistics serif::mapping::DomainMapper::get_geometry_cache_statistics () const [nodiscard]

10.151.2.17 is_compactified_element()

bool serif::mapping::DomainMapper::is_compactified_element (
const mfem::ElementTransformation & transformation) const [nodiscard], [noexcept]

10.151.2.18 is_geometry_cache_enabled()

bool serif::mapping::DomainMapper::is_geometry_cache_enabled () const [nodiscard], [noexcept]

10.151.2.19 operator=() [1/2]

DomainMapper & serif::mapping::DomainMapper::operator= (
const DomainMapper &) [delete]

10.151.2.20 operator=() [2/2]

DomainMapper & serif::mapping::DomainMapper::operator= (
DomainMapper &&) [default]

10.151.2.21 select_face_element_integration_point()

const mfem::IntegrationPoint & serif::mapping::DomainMapper::select_face_element_integration_point (
const mfem::FaceElementTransformations & transformations,
FaceElementSide side) [static], [nodiscard], [private]

10.151.2.22 select_face_element_transformation()

mfem::ElementTransformation & serif::mapping::DomainMapper::select_face_element_transformation (
const mfem::FaceElementTransformations & transformations,
FaceElementSide side) [static], [nodiscard], [private]

10.151.2.23 set_geometry_cache_enabled()

void serif::mapping::DomainMapper::set_geometry_cache_enabled (
bool enabled) const

10.151.2.24 validate_element_data()

void serif::mapping::DomainMapper::validate_element_data (
const ElementMappingData & element_data) const [private]

10.151.3 Member Data Documentation

10.151.3.1 m_exterior_map

std::unique_ptr<const compactification::ExteriorDomainMap> serif::mapping::DomainMapper::m_exterior_map [private]

10.151.3.2 m_geometryCache

std::unique_ptr<DMGeometryCache> serif::mapping::DomainMapper::m_geometryCache [mutable], [private]

10.151.3.3 m_options

utils::user_options::DomainMapperOptions serif::mapping::DomainMapper::m_options [private]

The documentation for this class was generated from the following files:

- src/include/serif/mapping/domain_mapper.hpp
- src/lib/serif/mapping/domain_mapper.cpp

10.152 serif::utils::user_options::DomainMapperOptions Struct Reference

#include <domain_mapper.hpp>

Public Attributes

- int dimension {3}
- int vacuum_element_attribute_ID {3}

10.152.1 Member Data Documentation

10.152.1.1 dimension

int serif::utils::user_options::DomainMapperOptions::dimension {3}

10.152.1.2 vacuum_element_attribute_ID

int serif::utils::user_options::DomainMapperOptions::vacuum_element_attribute_ID {3}

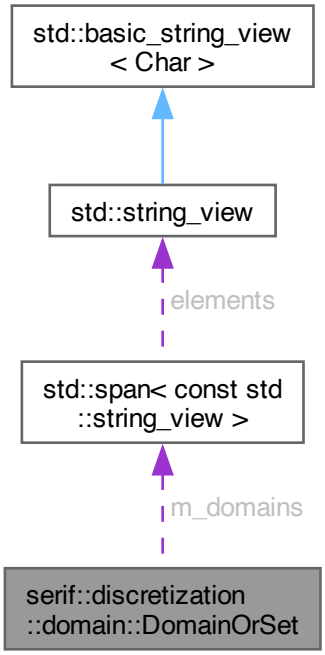
The documentation for this struct was generated from the following file:

- src/include/serif/utils/user_options/mapping/domain_mapper.hpp

10.153 serif::discretization::domain::DomainOrSet Class Reference

#include <concepts.hpp>

Collaboration diagram for serif::discretization::domain::DomainOrSet:



Public Member Functions

- template<IsDomain DomainT>
  constexpr DomainOrSet (DomainT) noexcept
- template<IsDomain... DomainTs>
  constexpr DomainOrSet (DomainSet< DomainTs... >) noexcept
- constexpr std::span< const std::string_view > domains () const noexcept

Private Attributes

- std::span< const std::string_view > m_domains

10.153.1 Constructor & Destructor Documentation

10.153.1.1 DomainOrSet() [1/2]

template<IsDomain DomainT>
serif::discretization::domain::DomainOrSet::DomainOrSet (
DomainT) [inline], [constexpr], [noexcept]

10.153.1.2 DomainOrSet() [2/2]

template<IsDomain... DomainTs>
serif::discretization::domain::DomainOrSet::DomainOrSet (
DomainSet< DomainTs... >) [inline], [constexpr], [noexcept]

10.153.2 Member Function Documentation

10.153.2.1 domains()

std::span< const std::string_view > serif::discretization::domain::DomainOrSet::domains () const [inline], [nodiscard], [constexpr], [noexcept]

10.153.3 Member Data Documentation

10.153.3.1 m_domains

std::span<const std::string_view> serif::discretization::domain::DomainOrSet::m_domains [private]

The documentation for this class was generated from the following file:

- src/include/serif/discretization/domain/concepts.hpp

10.154 serif::discretization::domain::DomainOrSetList< DomainTs > Struct Template Reference

#include <concepts.hpp>

Static Public Attributes

- static constexpr std::size_t count = CountVariadicArguments<DomainTs...>()

10.154.1 Member Data Documentation

10.154.1.1 count

template<IsDomainOrSet... DomainTs>
std::size_t serif::discretization::domain::DomainOrSetList< DomainTs >::count = CountVariadicArguments<DomainTs...>() [static], [constexpr]

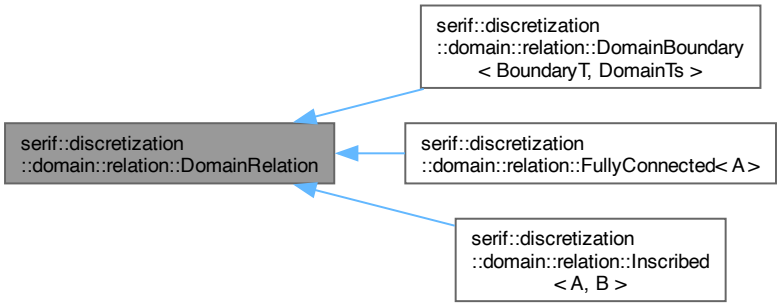
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/concepts.hpp

10.155 serif::discretization::domain::relation::DomainRelation Struct Reference

#include <relations.hpp>

Inheritance diagram for serif::discretization::domain::relation::DomainRelation:



The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/relations.hpp

10.156 serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations > Struct Template Reference

#include <domain_schema.hpp>

Public Types

- using domain_ids_type = DomainIDs
- using boundary_ids_type = BoundaryIDs
- using relations_type = Relations

Static Public Member Functions

- static constexpr auto domain_descriptors () noexcept
- static constexpr auto boundary_descriptors () noexcept
- template<IsDomainOrSet DomainT>
  static consteval bool contains_domain () noexcept
- template<IsDomainOrSet DomainT>
  static constexpr bool domain_id_belongs_to (const int domainID) noexcept
- template<IsDomain DomainT>
  static consteval int domain_id () noexcept
- template<IsBoundary BoundaryT>
  static consteval bool contains_boundary () noexcept
- template<IsBoundary BoundaryT>
  static consteval int boundary_id () noexcept
- template<IsBoundary BoundaryT>
  static constexpr bool boundary_id_matches (const int boundaryID) noexcept

Static Public Attributes

- static constexpr std::size_t domain_id_count = DomainIDs::count
- static constexpr std::size_t boundary_id_count = BoundaryIDs::count
- static constexpr std::size_t relation_count = Relations::count

10.156.1 Member Typedef Documentation

10.156.1.1 boundary_ids_type

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
using serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::boundary_ids_type = BoundaryIDs

10.156.1.2 domain_ids_type

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
using serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::domain_ids_type = DomainIDs

10.156.1.3 relations_type

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
using serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::relations_type = Relations

10.156.2 Member Function Documentation

10.156.2.1 boundary_descriptors()

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
static constexpr auto serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::boundary_descriptors () [inline], [static], [nodiscard], [constexpr], [noexcept]

10.156.2.2 boundary_id()

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
template<IsBoundary BoundaryT>
static consteval int serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::boundary_id () [inline], [static], [nodiscard], [consteval], [noexcept]

10.156.2.3 boundary_id_matches()

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
template<IsBoundary BoundaryT>
static constexpr bool serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::boundary_id_matches (
const int boundaryID) [inline], [static], [nodiscard], [constexpr], [noexcept]

10.156.2.4 contains_boundary()

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
template<IsBoundary BoundaryT>
static consteval bool serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::contains_boundary () [inline], [static], [nodiscard], [consteval], [noexcept]

10.156.2.5 contains_domain()

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
template<IsDomainOrSet DomainT>
static consteval bool serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::contains_domain () [inline], [static], [nodiscard], [consteval], [noexcept]

10.156.2.6 domain_descriptors()

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
static constexpr auto serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::domain_descriptors () [inline], [static], [nodiscard], [constexpr], [noexcept]

10.156.2.7 domain_id()

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
template<IsDomain DomainT>
static consteval int serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::domain_id () [inline], [static], [nodiscard], [consteval], [noexcept]

10.156.2.8 domain_id_belongs_to()

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
template<IsDomainOrSet DomainT>
static constexpr bool serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::domain_id_belongs_to (
const int domainID) [inline], [static], [nodiscard], [constexpr], [noexcept]

10.156.3 Member Data Documentation

10.156.3.1 boundary_id_count

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
std::size_t serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::boundary_id_count = BoundaryIDs::count [static], [constexpr]

10.156.3.2 domain_id_count

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
std::size_t serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::domain_id_count = DomainIDs::count [static], [constexpr]

10.156.3.3 relation_count

template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
std::size_t serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >::relation_count = Relations::count [static], [constexpr]

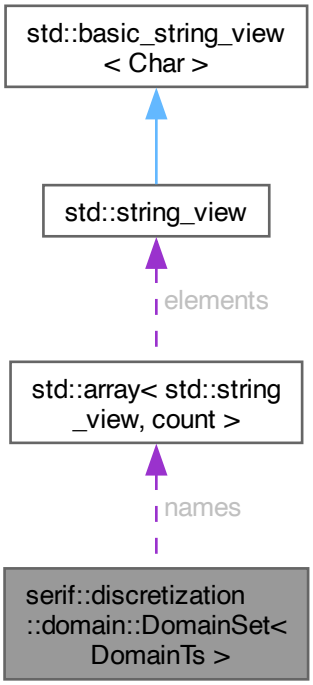
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/schema/domain_schema.hpp

10.157 serif::discretization::domain::DomainSet< DomainTs > Struct Template Reference

#include <concepts.hpp>

Collaboration diagram for serif::discretization::domain::DomainSet< DomainTs >:



Static Public Attributes

- static constexpr std::size_t count = CountVariadicArguments<DomainTs...>()
- static const std::array< std::string_view, count > names

10.157.1 Member Data Documentation

10.157.1.1 count

template<IsDomain... DomainTs>
std::size_t serif::discretization::domain::DomainSet< DomainTs >::count = CountVariadicArguments<DomainTs...>()
[static], [constexpr]

10.157.1.2 names

template<IsDomain... DomainTs>
const std::array<std::string_view, count> serif::discretization::domain::DomainSet< DomainTs >::names [inline], [static]

Initial value:
```
{
        DomainTs::name...
    }
```

The documentation for this struct was generated from the following file:

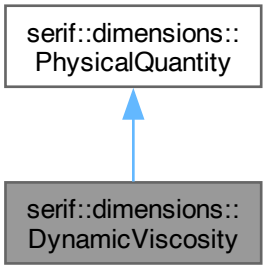
- src/include/serif/discretization/domain/concepts.hpp

10.158 serif::dimensions::DynamicViscosity Struct Reference

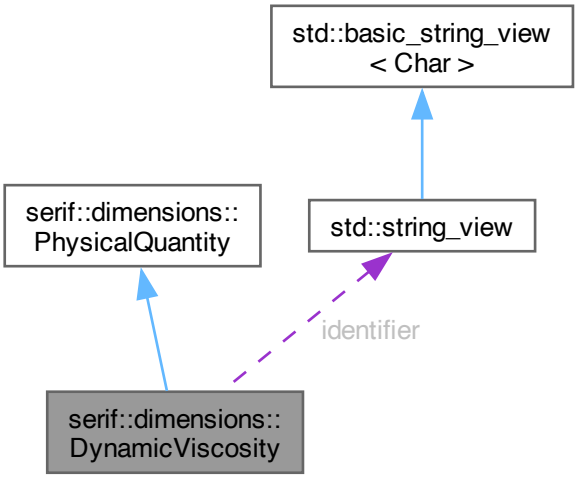
Dimension of Dynamic Viscosity.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::DynamicViscosity:



Collaboration diagram for serif::dimensions::DynamicViscosity:



Static Public Attributes

- static constexpr std::string_view identifier = ”dynamic_viscosity”

10.158.1 Detailed Description

Dimension of Dynamic Viscosity.

10.158.2 Member Data Documentation

10.158.2.1 identifier

std::string_view serif::dimensions::DynamicViscosity::identifier = ”dynamic_viscosity” [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.159 serif::operators::backend::mfem::ElementBasis< OperandCount > Struct Template Reference

Fixed-mesh basis samples; independent of the current coefficient values and displacement.

#include <element_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::ElementBasis< OperandCount >:



Public Attributes

- int element {-1}
- const ::mfem::IntegrationRule * rule {nullptr}
- BasisEvaluation reference_test_basis
- std::array< BasisEvaluation, OperandCount > reference_operand_basis
- BasisEvaluation displacement_basis

10.159.1 Detailed Description

template<std::size_t OperandCount>
struct serif::operators::backend::mfem::ElementBasis< OperandCount >

Fixed-mesh basis samples; independent of the current coefficient values and displacement.

10.159.2 Member Data Documentation

10.159.2.1 displacement_basis

template<std::size_t OperandCount>
BasisEvaluation serif::operators::backend::mfem::ElementBasis< OperandCount >::displacement_basis

10.159.2.2 element

template<std::size_t OperandCount>
int serif::operators::backend::mfem::ElementBasis< OperandCount >::element {-1}

10.159.2.3 reference_operand_basis

template<std::size_t OperandCount>
std::array<BasisEvaluation, OperandCount> serif::operators::backend::mfem::ElementBasis< OperandCount >::reference_↩
operand_basis

10.159.2.4 reference_test_basis

template<std::size_t OperandCount>
BasisEvaluation serif::operators::backend::mfem::ElementBasis< OperandCount >::reference_test_basis

10.159.2.5 rule

template<std::size_t OperandCount>
const ::mfem::IntegrationRule* serif::operators::backend::mfem::ElementBasis< OperandCount >::rule {nullptr}

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/element_data.hpp

10.160 serif::mapping::ElementCompactificationData Class Reference

#include <data.hpp>

Public Member Functions

- ElementCompactificationData (const mfem::FiniteElement &element, const mfem::Vector &dofs)
- const mfem::FiniteElement & get_element () const noexcept
- const mfem::Vector & get_dofs () const noexcept
- int get_dof_count () const noexcept

Private Attributes

- const mfem::FiniteElement * m_element
- mfem::Vector m_dofs

10.160.1 Constructor & Destructor Documentation

10.160.1.1 ElementCompactificationData()

serif::mapping::ElementCompactificationData::ElementCompactificationData (
const mfem::FiniteElement & element,
const mfem::Vector & dofs)

10.160.2 Member Function Documentation

10.160.2.1 get_dof_count()

int serif::mapping::ElementCompactificationData::get_dof_count () const [nodiscard], [noexcept]

10.160.2.2 get_dofs()

const mfem::Vector & serif::mapping::ElementCompactificationData::get_dofs () const [nodiscard], [noexcept]

10.160.2.3 get_element()

const mfem::FiniteElement & serif::mapping::ElementCompactificationData::get_element () const [nodiscard], [noexcept]

10.160.3 Member Data Documentation

10.160.3.1 m_dofs

mfem::Vector serif::mapping::ElementCompactificationData::m_dofs [private]

10.160.3.2 m_element

const mfem::FiniteElement* serif::mapping::ElementCompactificationData::m_element [private]

The documentation for this class was generated from the following files:

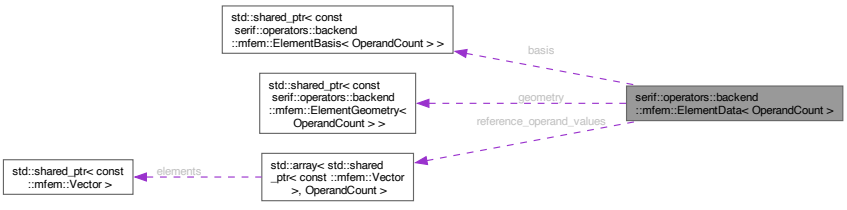
- src/include/serif/mapping/data.hpp
- src/lib/serif/mapping/data.cpp

10.161 serif::operators::backend::mfem::ElementData< OperandCount > Struct Template Reference

Weighted equation data retains the exact basis, geometry, and samples used to compute it.

#include <element_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::ElementData< OperandCount >:



Public Attributes

- int element {-1}
- std::shared_ptr< const ElementBasis< OperandCount > > basis
- std::shared_ptr< const ElementGeometry< OperandCount > > geometry
- std::array< std::shared_ptr< const ::mfem::Vector >, OperandCount > reference_operand_values
- ::mfem::Vector weighted_residual
- ::mfem::DenseMatrix weighted_field_derivatives
- ::mfem::DenseMatrix weighted_position_derivative
- double validation_integral {0.0}

10.161.1 Detailed Description

template<std::size_t OperandCount>
struct serif::operators::backend::mfem::ElementData< OperandCount >

Weighted equation data retains the exact basis, geometry, and samples used to compute it.

10.161.2 Member Data Documentation

10.161.2.1 basis

template<std::size_t OperandCount>
std::shared_ptr<const ElementBasis<OperandCount> > serif::operators::backend::mfem::ElementData< OperandCount >↩
::basis

10.161.2.2 element

template<std::size_t OperandCount>
int serif::operators::backend::mfem::ElementData< OperandCount >::element {-1}

10.161.2.3 geometry

template<std::size_t OperandCount>
std::shared_ptr<const ElementGeometry<OperandCount> > serif::operators::backend::mfem::ElementData< OperandCount
>::geometry

10.161.2.4 reference_operand_values

template<std::size_t OperandCount>
std::array<std::shared_ptr<const ::mfem::Vector>, OperandCount> serif::operators::backend::mfem::ElementData< Operand↩
Count >::reference_operand_values

10.161.2.5 validation_integral

template<std::size_t OperandCount>
double serif::operators::backend::mfem::ElementData< OperandCount >::validation_integral {0.0}

10.161.2.6 weighted_field_derivatives

template<std::size_t OperandCount>
::mfem::DenseMatrix serif::operators::backend::mfem::ElementData< OperandCount >::weighted_field_derivatives

10.161.2.7 weighted_position_derivative

template<std::size_t OperandCount>
::mfem::DenseMatrix serif::operators::backend::mfem::ElementData< OperandCount >::weighted_position_derivative

10.161.2.8 weighted_residual

template<std::size_t OperandCount>
::mfem::Vector serif::operators::backend::mfem::ElementData< OperandCount >::weighted_residual

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/element_data.hpp

10.162 serif::mapping::ElementDisplacementData Class Reference

#include <data.hpp>

Public Member Functions

- ElementDisplacementData (const mfem::FiniteElement &element, mfem::Vector &displacement_dofs, mfem::Ordering::Type ordering=mfem::Ordering::byNODES)
- const mfem::FiniteElement & get_element () const noexcept
- const mfem::DenseMatrix & get_dof_matrix () const noexcept
- int get_dimension () const noexcept
- int get_dof_count () const noexcept
- mfem::Ordering::Type get_ordering () const noexcept

Private Attributes

- const mfem::FiniteElement * m_element
- mfem::DenseMatrix m_dof_matrix
- int m_dimension
- mfem::Ordering::Type m_ordering

10.162.1 Constructor & Destructor Documentation

10.162.1.1 ElementDisplacementData()

serif::mapping::ElementDisplacementData::ElementDisplacementData (
const mfem::FiniteElement & element,
mfem::Vector & displacement_dofs,
mfem::Ordering::Type ordering = mfem::Ordering::byNODES)

10.162.2 Member Function Documentation

10.162.2.1 get_dimension()

int serif::mapping::ElementDisplacementData::get_dimension () const [nodiscard], [noexcept]

10.162.2.2 get_dof_count()

int serif::mapping::ElementDisplacementData::get_dof_count () const [nodiscard], [noexcept]

10.162.2.3 get_dof_matrix()

const mfem::DenseMatrix & serif::mapping::ElementDisplacementData::get_dof_matrix () const [nodiscard], [noexcept]

10.162.2.4 get_element()

const mfem::FiniteElement & serif::mapping::ElementDisplacementData::get_element () const [nodiscard], [noexcept]

10.162.2.5 get_ordering()

Ordering::Type serif::mapping::ElementDisplacementData::get_ordering () const [nodiscard], [noexcept]

10.162.3 Member Data Documentation

10.162.3.1 m_dimension

int serif::mapping::ElementDisplacementData::m_dimension [private]

10.162.3.2 m_dof_matrix

mfem::DenseMatrix serif::mapping::ElementDisplacementData::m_dof_matrix [private]

10.162.3.3 m_element

const mfem::FiniteElement* serif::mapping::ElementDisplacementData::m_element [private]

10.162.3.4 m_ordering

mfem::Ordering::Type serif::mapping::ElementDisplacementData::m_ordering [private]

The documentation for this class was generated from the following files:

- src/include/serif/mapping/data.hpp
- src/lib/serif/mapping/data.cpp

10.163 serif::operators::backend::mfem::ElementGeometry< OperandCount > Struct Template Reference

Physical mapping and mapped bases for one frozen displacement.

#include <element_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::ElementGeometry< OperandCount >:



Public Attributes

- BasisEvaluation test_basis
- std::array< BasisEvaluation, OperandCount > operand_basis
- ::mfem::Vector displacement
- ::mfem::Vector compactification
- mapping::VolumeMappingCache mapping

10.163.1 Detailed Description

template<std::size_t OperandCount>
struct serif::operators::backend::mfem::ElementGeometry< OperandCount >

Physical mapping and mapped bases for one frozen displacement.

10.163.2 Member Data Documentation

10.163.2.1 compactification

template<std::size_t OperandCount>
::mfem::Vector serif::operators::backend::mfem::ElementGeometry< OperandCount >::compactification

10.163.2.2 displacement

template<std::size_t OperandCount>
::mfem::Vector serif::operators::backend::mfem::ElementGeometry< OperandCount >::displacement

10.163.2.3 mapping

template<std::size_t OperandCount>
mapping::VolumeMappingCache serif::operators::backend::mfem::ElementGeometry< OperandCount >::mapping

10.163.2.4 operand_basis

template<std::size_t OperandCount>
std::array<BasisEvaluation, OperandCount> serif::operators::backend::mfem::ElementGeometry< OperandCount >::operand↩
_basis

10.163.2.5 test_basis

template<std::size_t OperandCount>
BasisEvaluation serif::operators::backend::mfem::ElementGeometry< OperandCount >::test_basis

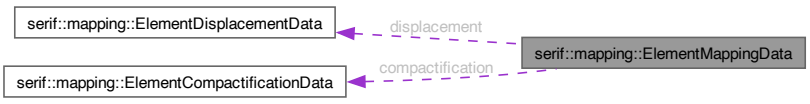
The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/element_data.hpp

10.164 serif::mapping::ElementMappingData Struct Reference

#include <data.hpp>

Collaboration diagram for serif::mapping::ElementMappingData:



Public Attributes

- const ElementDisplacementData & displacement
- const ElementCompactificationData & compactification

10.164.1 Member Data Documentation

10.164.1.1 compactification

const ElementCompactificationData& serif::mapping::ElementMappingData::compactification

10.164.1.2 displacement

const ElementDisplacementData& serif::mapping::ElementMappingData::displacement

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/data.hpp

10.165 serif::geometry::deformation::EmptyDeformationWorkspace Struct Reference

Stateless linear actions need no hidden mutable scratch.

#include <actions.hpp>

10.165.1 Detailed Description

Stateless linear actions need no hidden mutable scratch.

The documentation for this struct was generated from the following file:

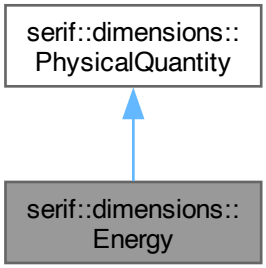
- src/include/serif/geometry/deformation/actions.hpp

10.166 serif::dimensions::Energy Struct Reference

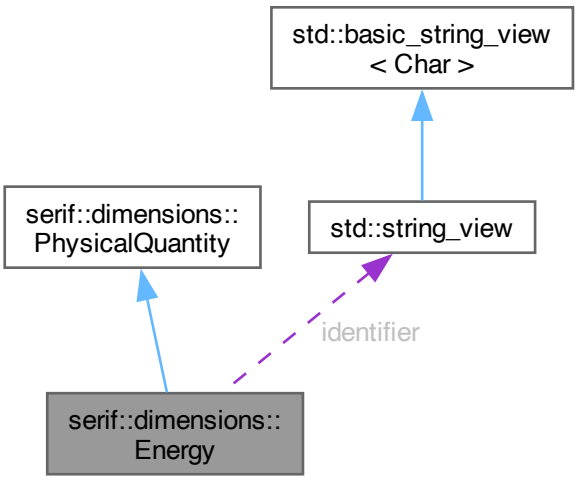
Dimension of Energy.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Energy:



Collaboration diagram for serif::dimensions::Energy:



Static Public Attributes

- static constexpr std::string_view identifier = "energy"

10.166.1 Detailed Description

Dimension of Energy.

10.166.2 Member Data Documentation

10.166.2.1 identifier

std::string_view serif::dimensions::Energy::identifier = "energy" [static], [constexpr]

The documentation for this struct was generated from the following file:

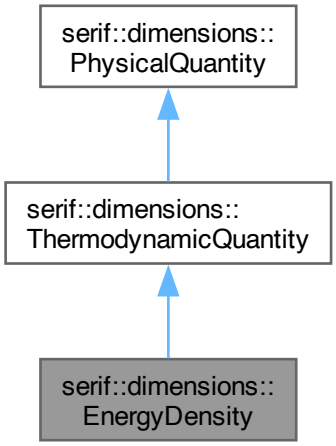
- src/include/serif/dimensions/dimensions.hpp

10.167 serif::dimensions::EnergyDensity Struct Reference

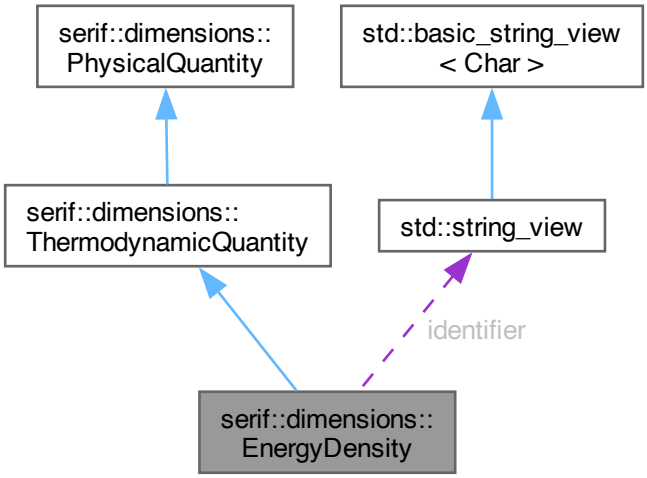
Dimension of Energy Density.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::EnergyDensity:



Collaboration diagram for serif::dimensions::EnergyDensity:



Static Public Attributes

- static constexpr std::string_view identifier = "energy_density"

10.167.1 Detailed Description

Dimension of Energy Density.

10.167.2 Member Data Documentation

10.167.2.1 identifier

std::string_view serif::dimensions::EnergyDensity::identifier = "energy_density" [static], [constexpr]

The documentation for this struct was generated from the following file:

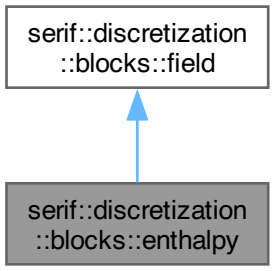
- src/include/serif/dimensions/dimensions.hpp

10.168 serif::discretization::blocks::enthalpy Struct Reference

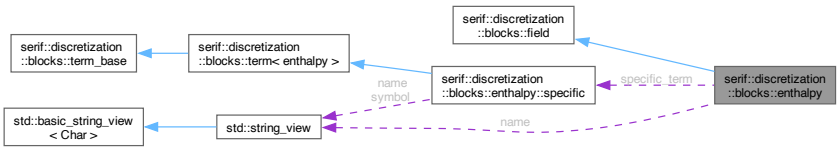
The enthalpy field is a scalar field is a C0 continuous scalar field. Note that this field likely will not survive the transition to the more general non-barotropic case. In the barotropic case the gradient of enthalpy is parallel to the gradient of the pressure. This make enthalpy a particularly useful quantity as we can represent the 0 pressure surface condition as the 0 enthalpy condition and we can break down mass constraint into a game of optimizing the Bernoulli constant.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::enthalpy:



Collaboration diagram for serif::discretization::blocks::enthalpy:



Classes

- struct specific

  the specific enthalpy is the term we specifically work with rather than the total enthalpy of dof. In fact heuristically I would say that generally we should always work in specific quantities rather than total quantities as there is not a well defined amount of material at some dof.

Public Types

- using Terms = utils::types::TypeList<specific>

  Type list of terms for compile time reflection and iteration over the terms of the field.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<enthalpy>()
  Compile time reflected name of the field.
- static constexpr specific specific_term {}
  Static constexpr instance of the specific term. This is used to access the term in the discretization.

10.168.1 Detailed Description

The enthalpy field is a scalar field is a C0 continuous scalar field. Note that this field likely will not survive the transition to the more general non-barotropic case. In the barotropic case the gradient of enthalpy is parallel to the gradient of the pressure. This make enthalpy a particularly useful quantity as we can represent the 0 pressure surface condition as the 0 enthalpy condition and we can break down mass constraint into a game of optimizing the Bernoulli constant.

10.168.2 Member Typedef Documentation

10.168.2.1 Terms

using serif::discretization::blocks::enthalpy::Terms = utils::types::TypeList<specific>

Type list of terms for compile time reflection and iteration over the terms of the field.

10.168.3 Member Data Documentation

10.168.3.1 name

std::string_view serif::discretization::blocks::enthalpy::name = meta::type_name<enthalpy>() [static], [constexpr]

Compile time reflected name of the field.

10.168.3.2 specific_term

specific serif::discretization::blocks::enthalpy::specific_term {} [static], [constexpr]

Static constexpr instance of the specific term. This is used to access the term in the discretization.

The documentation for this struct was generated from the following file:

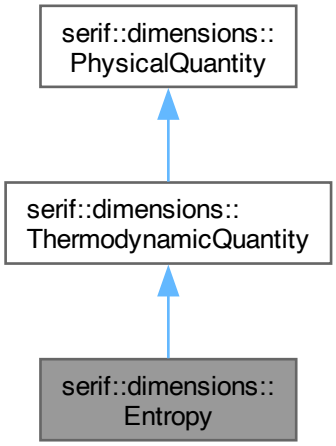
- src/include/serif/discretization/blocks/fields.hpp

10.169 serif::dimensions::Entropy Struct Reference

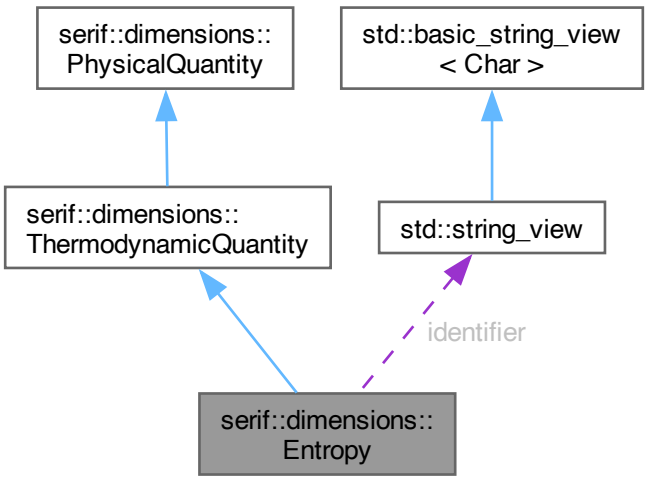
Dimension of Entropy.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Entropy:



Collaboration diagram for serif::dimensions::Entropy:



Static Public Attributes

- static constexpr std::string_view identifier = ”entropy”

10.169.1 Detailed Description

Dimension of Entropy.

10.169.2 Member Data Documentation

10.169.2.1 identifier

std::string_view serif::dimensions::Entropy::identifier = ”entropy” [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.170 serif::discretization::backend::mfem::TensorBasisCache::Entry Struct Reference

A struct that represents an entry in the TensorBasisCache. Each entry contains a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

Collaboration diagram for serif::discretization::backend::mfem::TensorBasisCache::Entry:



Public Attributes

- const ::mfem::FiniteElement * element
- ReferenceOperation operation
- int copies
- std::vector< std::array< double, 4 > > points
- std::shared_ptr< const TensorBasis > basis

10.170.1 Detailed Description

A struct that represents an entry in the TensorBasisCache. Each entry contains a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

Todo Consider if we can remove the shared pointer from here, shared pointers introduce overhead with reference counting and that may slow down the cache.

10.170.2 Member Data Documentation

10.170.2.1 basis

std::shared_ptr<const TensorBasis> serif::discretization::backend::mfem::TensorBasisCache::Entry::basis

10.170.2.2 copies

int serif::discretization::backend::mfem::TensorBasisCache::Entry::copies

10.170.2.3 element

const ::mfem::FiniteElement* serif::discretization::backend::mfem::TensorBasisCache::Entry::element

10.170.2.4 operation

ReferenceOperation serif::discretization::backend::mfem::TensorBasisCache::Entry::operation

10.170.2.5 points

std::vector<std::array<double, 4> > serif::discretization::backend::mfem::TensorBasisCache::Entry::points

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/backend/mfem/tensor_basis.hpp

10.171 serif::operators::backend::mfem::TensorBasisCache::Entry Struct Reference

A struct that represents an entry in the TensorBasisCache. Each entry contains a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

Collaboration diagram for serif::operators::backend::mfem::TensorBasisCache::Entry:



Public Attributes

- const ::mfem::FiniteElement * element
- ReferenceOperation operation
- int copies
- std::vector< std::array< double, 4 > > points
- std::shared_ptr< const TensorBasis > basis

10.171.1 Detailed Description

A struct that represents an entry in the TensorBasisCache. Each entry contains a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

Todo Consider if we can remove the shared pointer from here, shared pointers introduce overhead with reference counting and that may slow down the cache.

10.171.2 Member Data Documentation

10.171.2.1 basis

std::shared_ptr<const TensorBasis> serif::discretization::backend::mfem::TensorBasisCache::Entry::basis

10.171.2.2 copies

int serif::discretization::backend::mfem::TensorBasisCache::Entry::copies

10.171.2.3 element

const ::mfem::FiniteElement* serif::discretization::backend::mfem::TensorBasisCache::Entry::element

10.171.2.4 operation

ReferenceOperation serif::discretization::backend::mfem::TensorBasisCache::Entry::operation

10.171.2.5 points

std::vector<std::array<double, 4> > serif::discretization::backend::mfem::TensorBasisCache::Entry::points

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/backend/mfem/tensor_basis.hpp

10.172 serif::utils::misc::concepts::EnumerateTypes< TypeList, Descriptor > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/enumeration.hpp

10.173 serif::utils::misc::concepts::EnumerateTypes< types::TypeList< TypeTs... >, Descriptor > Struct Template Reference

#include <enumeration.hpp>

Public Types

- using Type = ConcatT<Descriptor<TypeTs>...>

10.173.1 Member Typedef Documentation

10.173.1.1 Type

template<template< typename > typename Descriptor, typename... TypeTs>
using serif::utils::misc::concepts::EnumerateTypes< types::TypeList< TypeTs... >, Descriptor >::Type = ConcatT<Descriptor<TypeTs>...>

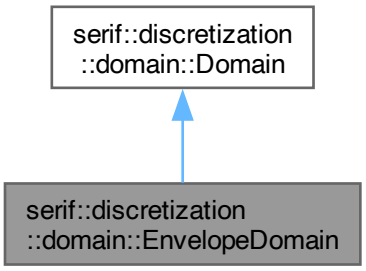
The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/enumeration.hpp

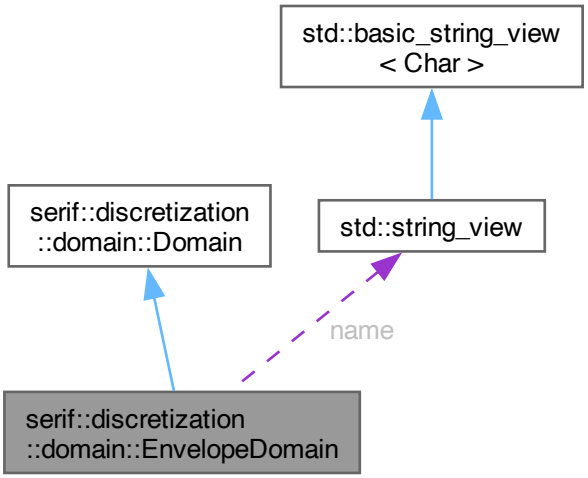
10.174 serif::discretization::domain::EnvelopeDomain Struct Reference

#include <types.hpp>

Inheritance diagram for serif::discretization::domain::EnvelopeDomain:



Collaboration diagram for serif::discretization::domain::EnvelopeDomain:



Static Public Attributes

- static constexpr std::string_view name = "envelope"

10.174.1 Member Data Documentation

10.174.1.1 name

std::string_view serif::discretization::domain::EnvelopeDomain::name = ”envelope” [static], [constexpr]

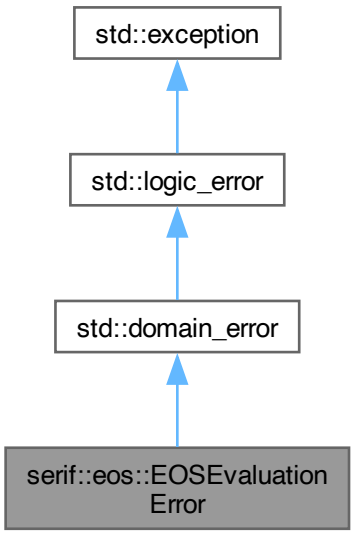
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/types.hpp

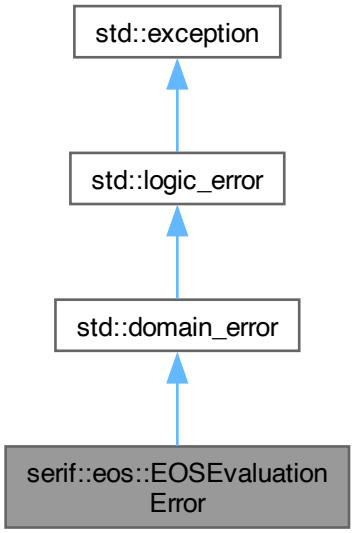
10.175 serif::eos::EOSEvaluationError Class Reference

#include <exceptions.hpp>

Inheritance diagram for serif::eos::EOSEvaluationError:



Collaboration diagram for serif::eos::EOSEvaluationError:



Public Member Functions

- EOSEvaluationError (std::string message)

10.175.1 Constructor & Destructor Documentation

10.175.1.1 EOSEvaluationError()

serif::eos::EOSEvaluationError::EOSEvaluationError (
std::string message) [inline], [explicit]

The documentation for this class was generated from the following file:

- src/include/serif/eos/exceptions.hpp

10.176 serif::geometry::surface::keys::EOSResponse Struct Reference

#include <samples.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/samples.hpp

10.177 serif::eos::runtime::EOSView Class Reference

#include <views.hpp>

Collaboration diagram for serif::eos::runtime::EOSView:



Public Member Functions

- template<typename EOS>
  requires RuntimeEOSModel<std::remove_cvref_t<EOS>>
  EOSView (const EOS &eos) noexcept
- template<typename EOS>
  requires RuntimeEOSModel<std::remove_cvref_t<EOS>>
  EOSView (const EOS &&)=delete
- std::span< const dimensions::runtime::RuntimeRelationDescriptor > relations () const noexcept
- bool supports (const dimensions::runtime::ThermodynamicQuantityID outputQuantity, const std::span< const dimensions::runtime::ThermodynamicQuantityID > inputQuantities) const noexcept
- std::expected< dimensions::runtime::RuntimeQuantityValue, utils::errors::SERiFErrorCode > try_evaluate (const dimensions::runtime::ThermodynamicQuantityID outputQuantity, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputQuantities) const
- template<dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity OutputQuantity, dimensions::ThermodynamicQuantityValueType... InputValues>
  std::expected< dimensions::QuantityValue< OutputQuantity >, utils::errors::SERiFErrorCode > try_evaluate (const InputValues... inputValues) const
- std::expected< double, utils::errors::SERiFErrorCode > try_partial_derivative (const dimensions::runtime::Thermodyn
  outputQuantity, const dimensions::runtime::ThermodynamicQuantityID withRespectTo, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues) const
- template<dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity OutputQuantity, dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity InputQuantity, dimensions::QuantityValueType... InputValues>
  std::expected< dimensions::PartialDerivative< OutputQuantity, InputQuantity >, utils::errors::SERiFErrorCode > try_partial_derivative (const InputValues... inputValues) const

Private Types

- using RuntimeEvaluateFunction
- using RuntimePartialDerivativeFunction

Private Member Functions

- const dimensions::runtime::RuntimeRelationDescriptor * find_relation (const dimensions::runtime::ThermodynamicQua
  outputQuantity, const std::span< const dimensions::runtime::ThermodynamicQuantityID > inputQuantities) const noexcept
- std::expected< const dimensions::runtime::RuntimeRelationDescriptor *, utils::errors::SERiFErrorCode > validate_relation_request (const dimensions::runtime::ThermodynamicQuantityID outputQuantity, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues) const

Static Private Member Functions

- template<typename Value>
  static std::expected< Value, utils::errors::SERiFErrorCode > runtime_failure (const utils::errors::SERiFErrorCode code)

Private Attributes

- const void * m_object
- std::span< const dimensions::runtime::RuntimeRelationDescriptor > m_relations
- RuntimeEvaluateFunction m_evaluate
- RuntimePartialDerivativeFunction m_partialDerivative

10.177.1 Member Typedef Documentation

10.177.1.1 RuntimeEvaluateFunction

using serif::eos::runtime::EOSView::RuntimeEvaluateFunction [private]

Initial value:

```
std::expected<double, utils::errors::SERiFErrorCode> (*)(
        const void*,
        const dimensions::runtime::ThermodynamicQuantityID&,
        std::span<const dimensions::runtime::RuntimeQuantityValue>
    )
```

10.177.1.2 RuntimePartialDerivativeFunction

using serif::eos::runtime::EOSView::RuntimePartialDerivativeFunction [private]

Initial value:

```
std::expected<double, utils::errors::SERiFErrorCode> (*)(
        const void*,
        const dimensions::runtime::ThermodynamicQuantityID&,
        const dimensions::runtime::ThermodynamicQuantityID&,
        std::span<const dimensions::runtime::RuntimeQuantityValue>
    )
```

10.177.2 Constructor & Destructor Documentation

10.177.2.1 EOSView() [1/2]

template<typename EOS>
requires RuntimeEOSModel<std::remove_cvref_t<EOS>>
serif::eos::runtime::EOSView::EOSView (
const EOS & eos) [inline], [explicit], [noexcept]

10.177.2.2 EOSView() [2/2]

template<typename EOS>
requires RuntimeEOSModel<std::remove_cvref_t<EOS>>
serif::eos::runtime::EOSView::EOSView (
const EOS &&) [explicit], [delete]

10.177.3 Member Function Documentation

10.177.3.1 find_relation()

const dimensions::runtime::RuntimeRelationDescriptor * serif::eos::runtime::EOSView::find_relation (
const dimensions::runtime::ThermodynamicQuantityID outputQuantity,
const std::span< const dimensions::runtime::ThermodynamicQuantityID > inputQuantities) const [inline], [nodiscard], [private], [noexcept]

10.177.3.2 relations()

std::span< const dimensions::runtime::RuntimeRelationDescriptor > serif::eos::runtime::EOSView::relations () const [inline], [nodiscard], [noexcept]

10.177.3.3 runtime_failure()

template<typename Value>
static std::expected< Value, utils::errors::SERiFErrorCode > serif::eos::runtime::EOSView::runtime_failure (
const utils::errors::SERiFErrorCode code) [inline], [static], [nodiscard], [private]

10.177.3.4 supports()

bool serif::eos::runtime::EOSView::supports (
const dimensions::runtime::ThermodynamicQuantityID outputQuantity,
const std::span< const dimensions::runtime::ThermodynamicQuantityID > inputQuantities) const [inline], [nodiscard], [noexcept]

10.177.3.5 try_evaluate() [1/2]

std::expected< dimensions::runtime::RuntimeQuantityValue, utils::errors::SERiFErrorCode > serif::eos::runtime::EOSView↩
::try_evaluate (
const dimensions::runtime::ThermodynamicQuantityID outputQuantity,
const std::span< const dimensions::runtime::RuntimeQuantityValue > inputQuantities) const [inline], [nodiscard]

10.177.3.6 try_evaluate() [2/2]

template<dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity OutputQuantity, dimensions::ThermodynamicQuantityValueType... InputValues>
std::expected< dimensions::QuantityValue< OutputQuantity >, utils::errors::SERiFErrorCode > serif::eos::runtime::EOSView::try_evaluate (
const InputValues... inputValues) const [inline], [nodiscard]

10.177.3.7 try_partial_derivative() [1/2]

std::expected< double, utils::errors::SERiFErrorCode > serif::eos::runtime::EOSView::try_partial_derivative (
const dimensions::runtime::ThermodynamicQuantityID outputQuantity,
const dimensions::runtime::ThermodynamicQuantityID withRespectTo,
const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues) const [inline], [nodiscard]

10.177.3.8 try_partial_derivative() [2/2]

template<dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity OutputQuantity, dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity InputQuantity, dimensions::QuantityValueType... InputValues>
std::expected< dimensions::PartialDerivative< OutputQuantity, InputQuantity >, utils::errors::SERiFErrorCode > serif::eos::runtime::EOSView::try_partial_derivative (
const InputValues... inputValues) const [inline], [nodiscard]

10.177.3.9 validate_relation_request()

std::expected< const dimensions::runtime::RuntimeRelationDescriptor *, utils::errors::SERiFErrorCode > serif::eos::runtime::EOSView::validate_relation_request (
const dimensions::runtime::ThermodynamicQuantityID outputQuantity,
const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues) const [inline], [nodiscard], [private]

10.177.4 Member Data Documentation

10.177.4.1 m_evaluate

RuntimeEvaluateFunction serif::eos::runtime::EOSView::m_evaluate [private]

10.177.4.2 m_object

const void* serif::eos::runtime::EOSView::m_object [private]

10.177.4.3 m_partialDerivative

RuntimePartialDerivativeFunction serif::eos::runtime::EOSView::m_partialDerivative [private]

10.177.4.4 m_relations

std::span<const dimensions::runtime::RuntimeRelationDescriptor> serif::eos::runtime::EOSView::m_relations [private]

The documentation for this class was generated from the following file:

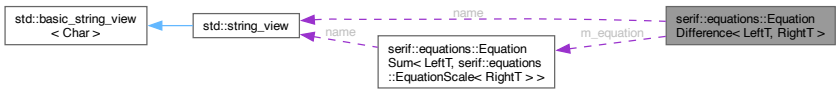
- src/include/serif/eos/runtime/views.hpp

10.178 serif::equations::EquationDifference< LeftT, RightT > Class Template Reference

Subtract every weak contribution of the right equation from the left equation.

#include <arithmetic.hpp>

Collaboration diagram for serif::equations::EquationDifference< LeftT, RightT >:



Public Types

- using Test = typename LeftT::Test

Public Member Functions

- EquationDifference (LeftT left, RightT right)
- const auto & contributions () const noexcept

Static Public Attributes

- static constexpr std::string_view name = "equation difference"

Private Attributes

- EquationSum< LeftT, EquationScale< RightT > > m_equation

10.178.1 Detailed Description

template<Equation LeftT, Equation RightT>
requires std::same_as<typename LeftT::Test, typename RightT::Test>
class serif::equations::EquationDifference< LeftT, RightT >

Subtract every weak contribution of the right equation from the left equation.

10.178.2 Member Typedef Documentation

10.178.2.1 Test

template<Equation LeftT, Equation RightT>
using serif::equations::EquationDifference< LeftT, RightT >::Test = typename LeftT::Test

10.178.3 Constructor & Destructor Documentation

10.178.3.1 EquationDifference()

template<Equation LeftT, Equation RightT>
serif::equations::EquationDifference< LeftT, RightT >::EquationDifference (
LeftT left,
RightT right) [inline], [explicit]

10.178.4 Member Function Documentation

10.178.4.1 contributions()

template<Equation LeftT, Equation RightT>
const auto & serif::equations::EquationDifference< LeftT, RightT >::contributions () const [inline], [nodiscard], [noexcept]

10.178.5 Member Data Documentation

10.178.5.1 m_equation

template<Equation LeftT, Equation RightT>
EquationSum<LeftT, EquationScale<RightT> > serif::equations::EquationDifference< LeftT, RightT >::m_equation [private]

10.178.5.2 name

template<Equation LeftT, Equation RightT>
std::string_view serif::equations::EquationDifference< LeftT, RightT >::name = "equation difference" [static], [constexpr]

The documentation for this class was generated from the following file:

- src/include/serif/equations/arithmetic.hpp

10.179 serif::geometry::surface::keys::EquationOfState Struct Reference

#include <samples.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/samples.hpp

10.180 serif::operators::EquationPlan< EquationT, OperandsT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

10.181 serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > > Struct Template Reference

#include <plan.hpp>

Public Types

- using Test = typename EquationT::Test
- using Support = discretization::blocks::SupportOf<Test>
- using Operands = utils::types::TypeList<O...>
- using Columns = typename detail::Columns<Operands>::type
- template<typename OperandT, typename ContributionT = EquationT>
  using Form = typename detail::CouplingForm<OperandT, equations::TestOperandOf<ContributionT>>↩
  ::type

Static Public Member Functions

- template<typename OperandT>
  static consteval std::size_t column_index ()

  Locate the unique coefficient column shared by every operation on a term.
- template<typename OperandT>
  static consteval int component_offset ()

  Locate the first scalar component of an operand in pointwise derivative storage.
- template<typename FunctionT>
  static void for_each_operand (FunctionT &&function)

  Visit all sampled operations in stable first-appearance order.
- template<typename ContributionT, typename FunctionT>
  static void for_each_input (FunctionT &&function)

  Visit only the operands needed by this contribution, retaining global sample indices.
- template<typename FunctionT>
  static void for_each_column (FunctionT &&function)

  Visit each physical coefficient vector once for binding and packed actions.

Static Public Attributes

- static constexpr std::size_t operand_count = sizeof...(O)
- static constexpr std::size_t column_count = equations::OperandValues<Columns>::count
- static constexpr int component_count = (discretization::forms::component_count<discretization::forms::OperandValue + ... + 0)

10.181.1 Member Typedef Documentation

10.181.1.1 Columns

template<VolumeEquation EquationT, typename... O>
using serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::Columns = typename detail::Columns<Operands>::type

10.181.1.2 Form

template<VolumeEquation EquationT, typename... O>
template<typename OperandT, typename ContributionT = EquationT>
using serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::Form = typename detail::CouplingForm<OperandT, equations::TestOperandOf<ContributionT>>::type

10.181.1.3 Operands

template<VolumeEquation EquationT, typename... O>
using serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::Operands = utils::types::TypeList<O...>

10.181.1.4 Support

template<VolumeEquation EquationT, typename... O>
using serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::Support = discretization::blocks::SupportOf<Test>

10.181.1.5 Test

template<VolumeEquation EquationT, typename... O>
using serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::Test = typename EquationT::Test

10.181.2 Member Function Documentation

10.181.2.1 column_index()

template<VolumeEquation EquationT, typename... O>
template<typename OperandT>
static consteval std::size_t serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::column_index () [inline], [static], [consteval]

Locate the unique coefficient column shared by every operation on a term.

10.181.2.2 component_offset()

template<VolumeEquation EquationT, typename... O>
template<typename OperandT>
static consteval int serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::component_offset () [inline], [static], [consteval]

Locate the first scalar component of an operand in pointwise derivative storage.

10.181.2.3 for_each_column()

template<VolumeEquation EquationT, typename... O>
template<typename FunctionT>
static void serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::for_each_column (
FunctionT && function) [inline], [static]

Visit each physical coefficient vector once for binding and packed actions.

10.181.2.4 for_each_input()

template<VolumeEquation EquationT, typename... O>
template<typename ContributionT, typename FunctionT>
static void serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::for_each_input (
FunctionT && function) [inline], [static]

Visit only the operands needed by this contribution, retaining global sample indices.

10.181.2.5 for_each_operand()

template<VolumeEquation EquationT, typename... O>
template<typename FunctionT>
static void serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::for_each_operand (
FunctionT && function) [inline], [static]

Visit all sampled operations in stable first-appearance order.

10.181.3 Member Data Documentation

10.181.3.1 column_count

template<VolumeEquation EquationT, typename... O>
std::size_t serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::column_count = equations::OperandValues<Columns>↩
::count [static], [constexpr]

10.181.3.2 component_count

template<VolumeEquation EquationT, typename... O>
int serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::component_count = (discretization::forms↩
::component_count<discretization::forms::OperandValue<O>> + ... + 0) [static], [constexpr]

10.181.3.3 operand_count

template<VolumeEquation EquationT, typename... O>
std::size_t serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >::operand_count = sizeof...(O) [static], [constexpr]

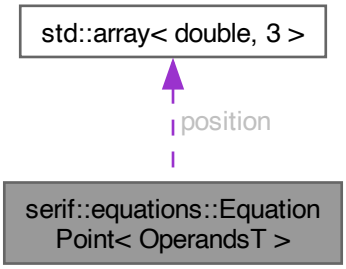
The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

10.182 serif::equations::EquationPoint< OperandsT > Struct Template Reference

#include <point.hpp>

Collaboration diagram for serif::equations::EquationPoint< OperandsT >:



Public Attributes

- OperandValues< OperandsT > fields
- std::array< double, 3 > position {}

10.182.1 Member Data Documentation

10.182.1.1 fields

template<typename OperandsT>
OperandValues<OperandsT> serif::equations::EquationPoint< OperandsT >::fields

10.182.1.2 position

template<typename OperandsT>
std::array<double, 3> serif::equations::EquationPoint< OperandsT >::position {}

The documentation for this struct was generated from the following file:

- src/include/serif/equations/point.hpp

10.183 serif::equations::EquationRegistry< EquationTs > Class Template Reference

#include <registry.hpp>

Public Types

- using Equations = utils::types::TypeList<EquationTs...>
- using Rows = utils::types::TypeList<discretization::forms::Operand<typename EquationTs::Test>...>

Public Member Functions

- EquationRegistry (EquationTs... equations)
- template<typename FunctionT>
  void for_each (FunctionT &&function) const
- template<Equation EquationT>
  const EquationT & get () const

Private Attributes

- std::tuple< EquationTs... > m_equations

10.183.1 Member Typedef Documentation

10.183.1.1 Equations

template<Equation... EquationTs>
using serif::equations::EquationRegistry< EquationTs >::Equations = utils::types::TypeList<EquationTs...>

10.183.1.2 Rows

template<Equation... EquationTs>
using serif::equations::EquationRegistry< EquationTs >::Rows = utils::types::TypeList<discretization::forms::Operand<typename EquationTs::Test>...>

10.183.2 Constructor & Destructor Documentation

10.183.2.1 EquationRegistry()

template<Equation... EquationTs>
serif::equations::EquationRegistry< EquationTs >::EquationRegistry (
EquationTs... equations) [inline], [explicit]

10.183.3 Member Function Documentation

10.183.3.1 for_each()

template<Equation... EquationTs>
template<typename FunctionT>
void serif::equations::EquationRegistry< EquationTs >::for_each (
FunctionT && function) const [inline]

10.183.3.2 get()

template<Equation... EquationTs>
template<Equation EquationT>
const EquationT & serif::equations::EquationRegistry< EquationTs >::get () const [inline], [nodiscard]

10.183.4 Member Data Documentation

10.183.4.1 m_equations

template<Equation... EquationTs>
std::tuple<EquationTs...> serif::equations::EquationRegistry< EquationTs >::m_equations [private]

The documentation for this class was generated from the following file:

- src/include/serif/equations/registry.hpp

10.184 serif::operators::EquationRegistry< EquationTs > Class Template Reference

#include <registry.hpp>

Public Types

- using Equations = utils::types::TypeList<EquationTs...>
- using Rows = utils::types::TypeList<discretization::forms::Operand<typename EquationTs::Test>...>

Public Member Functions

- EquationRegistry (EquationTs... equations)
- template<typename FunctionT>
  void for_each (FunctionT &&function) const
- template<Equation EquationT>
  const EquationT & get () const

Private Attributes

- std::tuple< EquationTs... > m_equations

10.184.1 Member Typedef Documentation

10.184.1.1 Equations

template<Equation... EquationTs>
using serif::equations::EquationRegistry< EquationTs >::Equations = utils::types::TypeList<EquationTs...>

10.184.1.2 Rows

template<Equation... EquationTs>
using serif::equations::EquationRegistry< EquationTs >::Rows = utils::types::TypeList<discretization::forms::Operand<typename EquationTs::Test>...>

10.184.2 Constructor & Destructor Documentation

10.184.2.1 EquationRegistry()

template<Equation... EquationTs>
serif::equations::EquationRegistry< EquationTs >::EquationRegistry (
EquationTs... equations) [inline], [explicit]

10.184.3 Member Function Documentation

10.184.3.1 for_each()

template<Equation... EquationTs>
template<typename FunctionT>
void serif::equations::EquationRegistry< EquationTs >::for_each (
FunctionT && function) const [inline]

10.184.3.2 get()

template<Equation... EquationTs>
template<Equation EquationT>
const EquationT & serif::equations::EquationRegistry< EquationTs >::get () const [inline], [nodiscard]

10.184.4 Member Data Documentation

10.184.4.1 m_equations

template<Equation... EquationTs>
std::tuple<EquationTs...> serif::equations::EquationRegistry< EquationTs >::m_equations [private]

The documentation for this class was generated from the following file:

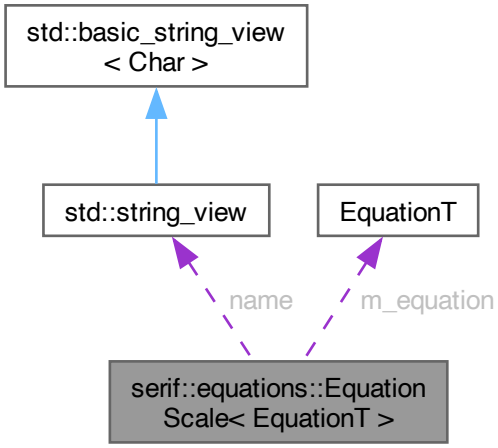
- src/include/serif/equations/registry.hpp

10.185 serif::equations::EquationScale< EquationT > Class Template Reference

Multiply an equation's weak contributions by a constant independent of state and position.

#include <arithmetic.hpp>

Collaboration diagram for serif::equations::EquationScale< EquationT >:



Public Types

- using Test = typename EquationT::Test

Public Member Functions

- EquationScale (EquationT equation, const double coefficient)
- auto contributions () const noexcept
- double coefficient () const noexcept

Static Public Attributes

- static constexpr std::string_view name = "scaled equation"

Private Attributes

- EquationT m_equation
- double m_coefficient

10.185.1 Detailed Description

template<Equation EquationT>
class serif::equations::EquationScale< EquationT >

Multiply an equation's weak contributions by a constant independent of state and position.

10.185.2 Member Typedef Documentation

10.185.2.1 Test

template<Equation EquationT>
using serif::equations::EquationScale< EquationT >::Test = typename EquationT::Test

10.185.3 Constructor & Destructor Documentation

10.185.3.1 EquationScale()

template<Equation EquationT>
serif::equations::EquationScale< EquationT >::EquationScale (
EquationT equation,
const double coefficient) [inline], [explicit]

10.185.4 Member Function Documentation

10.185.4.1 coefficient()

template<Equation EquationT>
double serif::equations::EquationScale< EquationT >::coefficient () const [inline], [nodiscard], [noexcept]

10.185.4.2 contributions()

template<Equation EquationT>
auto serif::equations::EquationScale< EquationT >::contributions () const [inline], [nodiscard], [noexcept]

10.185.5 Member Data Documentation

10.185.5.1 m_coefficient

template<Equation EquationT>
double serif::equations::EquationScale< EquationT >::m_coefficient [private]

10.185.5.2 m_equation

template<Equation EquationT>
EquationT serif::equations::EquationScale< EquationT >::m_equation [private]

10.185.5.3 name

template<Equation EquationT>
std::string_view serif::equations::EquationScale< EquationT >::name = "scaled equation" [static], [constexpr]

The documentation for this class was generated from the following file:

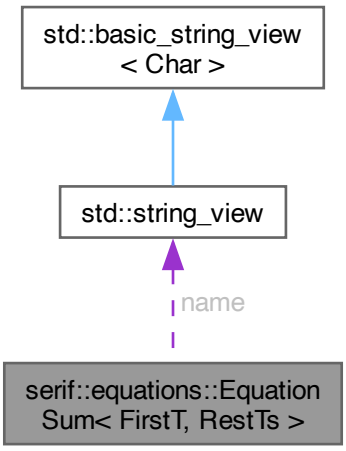
- src/include/serif/equations/arithmetic.hpp

10.186 serif::equations::EquationSum< FirstT, RestTs > Class Template Reference

Add equations sharing a residual row; each retains its own test operation and support.

#include <arithmetic.hpp>

Collaboration diagram for serif::equations::EquationSum< FirstT, RestTs >:



Public Types

- using Test = typename FirstT::Test

Public Member Functions

- EquationSum (FirstT first, RestTs... rest)
- const auto & contributions () const noexcept

Static Public Attributes

- static constexpr std::string_view name = "equation sum"

Private Attributes

- std::tuple< FirstT, RestTs... > m_equations

10.186.1 Detailed Description

template<Equation FirstT, Equation... RestTs>
requires (std::same_as<typename FirstT::Test, typename RestTs::Test> && ...)
class serif::equations::EquationSum< FirstT, RestTs >

Add equations sharing a residual row; each retains its own test operation and support.

10.186.2 Member Typedef Documentation

10.186.2.1 Test

template<Equation FirstT, Equation... RestTs>
using serif::equations::EquationSum< FirstT, RestTs >::Test = typename FirstT::Test

10.186.3 Constructor & Destructor Documentation

10.186.3.1 EquationSum()

template<Equation FirstT, Equation... RestTs>
serif::equations::EquationSum< FirstT, RestTs >::EquationSum (
FirstT first,
RestTs... rest) [inline], [explicit]

10.186.4 Member Function Documentation

10.186.4.1 contributions()

template<Equation FirstT, Equation... RestTs>
const auto & serif::equations::EquationSum< FirstT, RestTs >::contributions () const [inline], [nodiscard], [noexcept]

10.186.5 Member Data Documentation

10.186.5.1 m_equations

template<Equation FirstT, Equation... RestTs>
std::tuple<FirstT, RestTs...> serif::equations::EquationSum< FirstT, RestTs >::m_equations [private]

10.186.5.2 name

template<Equation FirstT, Equation... RestTs>
std::string_view serif::equations::EquationSum< FirstT, RestTs >::name = "equation sum" [static], [constexpr]

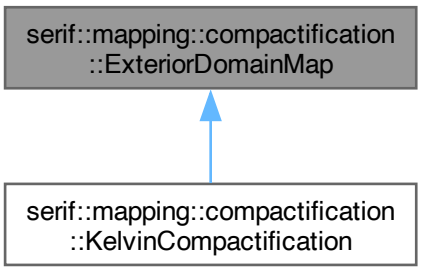
The documentation for this class was generated from the following file:

- src/include/serif/equations/arithmetic.hpp

10.187 serif::mapping::compactification::ExteriorDomainMap Class Reference

#include <compactification.hpp>

Inheritance diagram for serif::mapping::compactification::ExteriorDomainMap:



Public Member Functions

- virtual ~ExteriorDomainMap ()=default
- virtual bool is_affine_in_displacement () const noexcept=0
- virtual MappingStatus evaluate (const ExteriorMapInput &input, ExteriorMapResult &result) const =0
- virtual MappingStatus evaluate_variation (const ExteriorMapInput &input, const ExteriorMapResult &result, const ExteriorMapDirection &direction, ExteriorMapVariation &variation) const =0
- virtual std::string_view get_name () const noexcept=0
- virtual std::unique_ptr< const ExteriorDomainMap > clone () const =0

10.187.1 Constructor & Destructor Documentation

10.187.1.1 ~ExteriorDomainMap()

virtual serif::mapping::compactification::ExteriorDomainMap::~ExteriorDomainMap () [virtual], [default]

10.187.2 Member Function Documentation

10.187.2.1 clone()

virtual std::unique_ptr< const ExteriorDomainMap > serif::mapping::compactification::ExteriorDomainMap::clone () const [nodiscard], [pure virtual]

Implemented in serif::mapping::compactification::KelvinCompactification.

10.187.2.2 evaluate()

virtual MappingStatus serif::mapping::compactification::ExteriorDomainMap::evaluate (
const ExteriorMapInput & input,
ExteriorMapResult & result) const [nodiscard], [pure virtual]

Implemented in serif::mapping::compactification::KelvinCompactification.

10.187.2.3 evaluate_variation()

virtual MappingStatus serif::mapping::compactification::ExteriorDomainMap::evaluate_variation (
const ExteriorMapInput & input,
const ExteriorMapResult & result,
const ExteriorMapDirection & direction,
ExteriorMapVariation & variation) const [nodiscard], [pure virtual]

Implemented in serif::mapping::compactification::KelvinCompactification.

10.187.2.4 get_name()

virtual std::string_view serif::mapping::compactification::ExteriorDomainMap::get_name () const [nodiscard], [pure virtual], [noexcept]

Implemented in serif::mapping::compactification::KelvinCompactification.

10.187.2.5 is_affine_in_displacement()

virtual bool serif::mapping::compactification::ExteriorDomainMap::is_affine_in_displacement () const [nodiscard], [pure virtual], [noexcept]

Implemented in serif::mapping::compactification::KelvinCompactification.

The documentation for this class was generated from the following file:

- src/include/serif/mapping/compactification/compactification.hpp

10.188 serif::mapping::compactification::ExteriorMapDirection Struct Reference

#include <variation.hpp>

Public Attributes

- const mfem::Vector & displaced_position_variation
- const mfem::DenseMatrix & displacement_jacobian_variation

10.188.1 Member Data Documentation

10.188.1.1 displaced_position_variation

const mfem::Vector& serif::mapping::compactification::ExteriorMapDirection::displaced_position_variation

10.188.1.2 displacement_jacobian_variation

const mfem::DenseMatrix& serif::mapping::compactification::ExteriorMapDirection::displacement_jacobian_variation

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/compactification/variation.hpp

10.189 serif::mapping::compactification::ExteriorMapInput Struct Reference

#include <input_result.hpp>

Public Attributes

- const mfem::Vector & reference_position
- const mfem::Vector & displaced_position
- const mfem::DenseMatrix & displacement_jacobian
- const mfem::Vector & compactification_coordinate_gradient
- double compactification_coordinate

10.189.1 Member Data Documentation

10.189.1.1 compactification_coordinate

double serif::mapping::compactification::ExteriorMapInput::compactification_coordinate

10.189.1.2 compactification_coordinate_gradient

const mfem::Vector& serif::mapping::compactification::ExteriorMapInput::compactification_coordinate_gradient

10.189.1.3 displaced_position

const mfem::Vector& serif::mapping::compactification::ExteriorMapInput::displaced_position

10.189.1.4 displacement_jacobian

const mfem::DenseMatrix& serif::mapping::compactification::ExteriorMapInput::displacement_jacobian

10.189.1.5 reference_position

const mfem::Vector& serif::mapping::compactification::ExteriorMapInput::reference_position

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/compactification/input_result.hpp

10.190 serif::mapping::compactification::ExteriorMapResult Struct Reference

#include <input_result.hpp>

Public Attributes

- mfem::Vector physical_position
- mfem::DenseMatrix mapping_jacobian

10.190.1 Member Data Documentation

10.190.1.1 mapping_jacobian

mfem::DenseMatrix serif::mapping::compactification::ExteriorMapResult::mapping_jacobian

10.190.1.2 physical_position

mfem::Vector serif::mapping::compactification::ExteriorMapResult::physical_position

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/compactification/input_result.hpp

10.191 serif::mapping::compactification::ExteriorMapVariation Struct Reference

#include <variation.hpp>

Public Attributes

- mfem::Vector physical_position_variation
- mfem::DenseMatrix mapping_jacobian_variation

10.191.1 Member Data Documentation

10.191.1.1 mapping_jacobian_variation

mfem::DenseMatrix serif::mapping::compactification::ExteriorMapVariation::mapping_jacobian_variation

10.191.1.2 physical_position_variation

mfem::Vector serif::mapping::compactification::ExteriorMapVariation::physical_position_variation

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/compactification/variation.hpp

10.192 serif::discretization::domain::mesh::FaceElements Struct Reference

#include <topology.hpp>

Public Attributes

- int firstElementID {-1}
- int secondElementID {-1}

10.192.1 Member Data Documentation

10.192.1.1 firstElementID

int serif::discretization::domain::mesh::FaceElements::firstElementID {-1}

10.192.1.2 secondElementID

int serif::discretization::domain::mesh::FaceElements::secondElementID {-1}

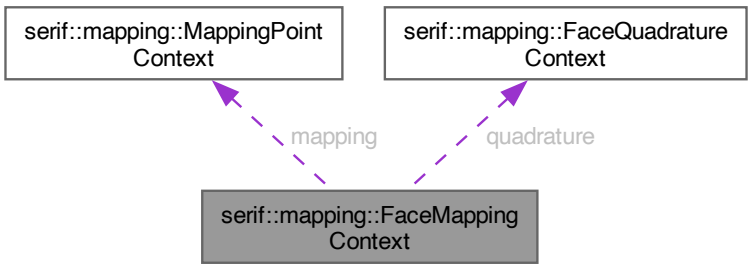
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/mesh/topology.hpp

10.193 serif::mapping::FaceMappingContext Struct Reference

#include <context.hpp>

Collaboration diagram for serif::mapping::FaceMappingContext:



Public Attributes

- MappingPointContext mapping
- FaceQuadratureContext quadrature
- mfem::Vector reference_normal
- double reference_surface_weight {0.0}
- double physical_surface_weight {0.0}

10.193.1 Member Data Documentation

10.193.1.1 mapping

MappingPointContext serif::mapping::FaceMappingContext::mapping

10.193.1.2 physical_surface_weight

double serif::mapping::FaceMappingContext::physical_surface_weight {0.0}

10.193.1.3 quadrature

FaceQuadratureContext serif::mapping::FaceMappingContext::quadrature

10.193.1.4 reference_normal

mfem::Vector serif::mapping::FaceMappingContext::reference_normal

10.193.1.5 reference_surface_weight

double serif::mapping::FaceMappingContext::reference_surface_weight {0.0}

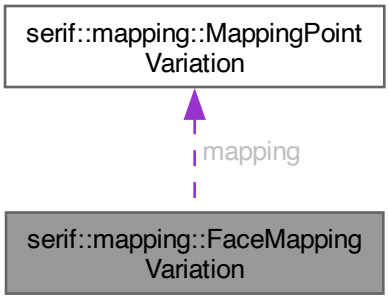
The documentation for this struct was generated from the following file:

- src/include/serif/mapping/context.hpp

10.194 serif::mapping::FaceMappingVariation Struct Reference

#include <variation.hpp>

Collaboration diagram for serif::mapping::FaceMappingVariation:



Public Attributes

- MappingPointVariation mapping
- mfem::Vector physical_normal_variation
- double physical_surface_weight_variation {0.0}
- double normal_flux_scale_variation {0.0}

10.194.1 Member Data Documentation

10.194.1.1 mapping

MappingPointVariation serif::mapping::FaceMappingVariation::mapping

10.194.1.2 normal_flux_scale_variation

double serif::mapping::FaceMappingVariation::normal_flux_scale_variation {0.0}

10.194.1.3 physical_normal_variation

mfem::Vector serif::mapping::FaceMappingVariation::physical_normal_variation

10.194.1.4 physical_surface_weight_variation

double serif::mapping::FaceMappingVariation::physical_surface_weight_variation {0.0}

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/variation.hpp

10.195 serif::mapping::FaceQuadratureContext Struct Reference

#include <context.hpp>

Public Attributes

- mfem::Vector normal
- double ds
- double v_dot_n_scale

10.195.1 Member Data Documentation

10.195.1.1 ds

double serif::mapping::FaceQuadratureContext::ds

10.195.1.2 normal

mfem::Vector serif::mapping::FaceQuadratureContext::normal

10.195.1.3 v_dot_n_scale

double serif::mapping::FaceQuadratureContext::v_dot_n_scale

The documentation for this struct was generated from the following file:

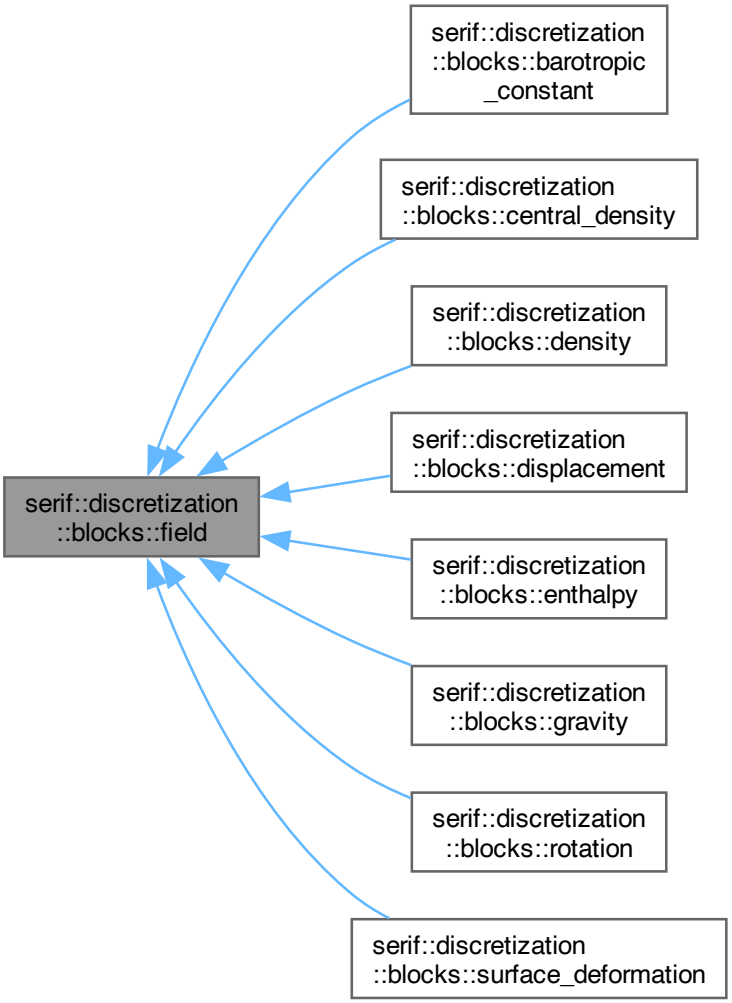
- src/include/serif/mapping/context.hpp

10.196 serif::discretization::blocks::field Struct Reference

The field struct is a tag type that is used to identify a field in the discretization. It is used to group together terms that belong to the same field. Each field should have a unique name and a list of terms that belong to it.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::field:



10.196.1 Detailed Description

The field struct is a tag type that is used to identify a field in the discretization. It is used to group together terms that belong to the same field. Each field should have a unique name and a list of terms that belong to it.

The documentation for this struct was generated from the following file:

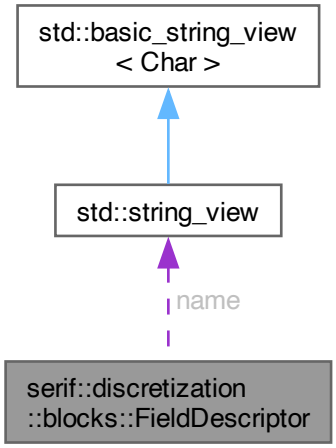
- src/include/serif/discretization/blocks/fields.hpp

10.197 serif::discretization::blocks::FieldDescriptor Struct Reference

Runtime descriptor for a field, containing its name. Two FieldDescriptors are equal if they have identical names.

#include <runtime.hpp>

Collaboration diagram for serif::discretization::blocks::FieldDescriptor:



Public Attributes

- std::string_view name
  The name of the field, used for runtime identification.

Friends

- constexpr bool operator== (const FieldDescriptor &, const FieldDescriptor &) noexcept=default
  Compare two FieldDescriptors for equality based on their names.

10.197.1 Detailed Description

Runtime descriptor for a field, containing its name. Two FieldDescriptors are equal if they have identical names.

10.197.2 Friends And Related Symbol Documentation

10.197.2.1 operator==

bool operator== (
const FieldDescriptor & ,
const FieldDescriptor &) [friend]

Compare two FieldDescriptors for equality based on their names.

Parameters

lhs	The left-hand side FieldDescriptor.
rhs	The right-hand side FieldDescriptor.

Returns

True if the names of both FieldDescriptors are equal, false otherwise.

10.197.3 Member Data Documentation

10.197.3.1 name

std::string_view serif::discretization::blocks::FieldDescriptor::name

The name of the field, used for runtime identification.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/runtime.hpp

10.198 serif::utils::misc::concepts::Filter< List, Predicate > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/filter.hpp

10.199 serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate > Struct Template Reference

#include <filter.hpp>

Public Types

- using Type = std::conditional_t<Predicate<Head>::value, PrependT<Head, FilteredTail>, FilteredTail>

Private Types

- using FilteredTail = Filter<types::TypeList<Tail...>, Predicate>::Type

10.199.1 Member Typedef Documentation

10.199.1.1 FilteredTail

template<typename Head, typename... Tail, template< typename > typename Predicate>
using serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate >::FilteredTail = Filter<types::TypeList<Tail...>, Predicate>::Type [private]

10.199.1.2 Type

template<typename Head, typename... Tail, template< typename > typename Predicate>
using serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate >::Type = std::conditional_t<Predicate<Head>::value, PrependT<Head, FilteredTail>, FilteredTail>

The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/filter.hpp

10.200 serif::utils::misc::concepts::Filter< types::TypeList<>, Predicate > Struct Template Reference

#include <filter.hpp>

Public Types

- using Type = types::TypeList<>

10.200.1 Member Typedef Documentation

10.200.1.1 Type

template<template< typename > typename Predicate>
using serif::utils::misc::concepts::Filter< types::TypeList<>, Predicate >::Type = types::TypeList<>

The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/filter.hpp

10.201 serif::geometry::deformation::FixedInfinityRadialExtension Struct Reference

Reduce surface motion linearly to zero at reference infinity in logical reference radius.

#include <models.hpp>

Public Member Functions

- bool operator== (const FixedInfinityRadialExtension &) const =default
  All fixed-infinity prescriptions have the same physical behavior.

10.201.1 Detailed Description

Reduce surface motion linearly to zero at reference infinity in logical reference radius.

10.201.2 Member Function Documentation

10.201.2.1 operator==()

bool serif::geometry::deformation::FixedInfinityRadialExtension::operator== (
const FixedInfinityRadialExtension &) const [nodiscard], [default]

All fixed-infinity prescriptions have the same physical behavior.

The documentation for this struct was generated from the following file:

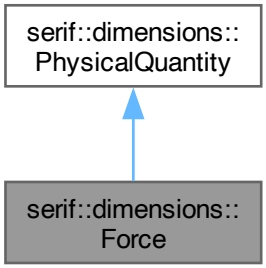
- src/include/serif/geometry/deformation/models.hpp

10.202 serif::dimensions::Force Struct Reference

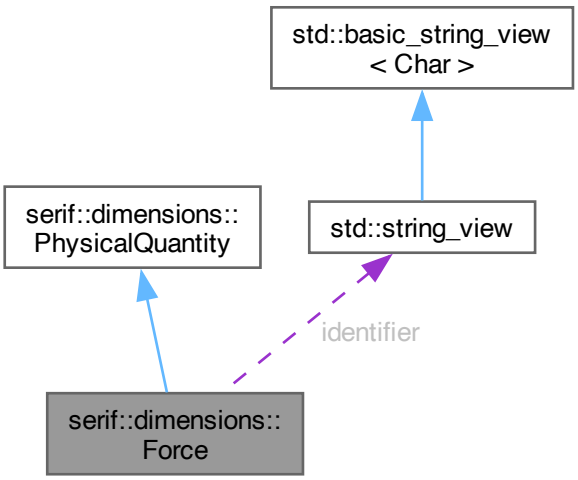
Dimension of Force.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Force:



Collaboration diagram for serif::dimensions::Force:



Static Public Attributes

- static constexpr std::string_view identifier = "force"

10.202.1 Detailed Description

Dimension of Force.

10.202.2 Member Data Documentation

10.202.2.1 identifier

std::string_view serif::dimensions::Force::identifier = "force" [static], [constexpr]

The documentation for this struct was generated from the following file:

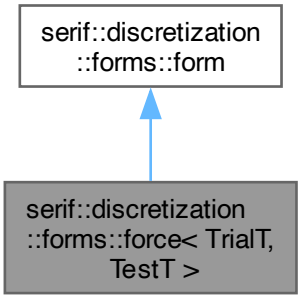
- src/include/serif/dimensions/dimensions.hpp

10.203 serif::discretization::forms::force< TrialT, TestT > Struct Template Reference

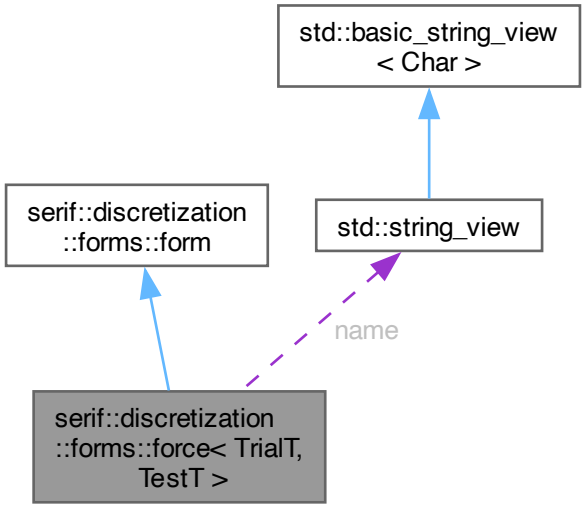
The force form represents the force term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the force operation.

#include <forms.hpp>

Inheritance diagram for serif::discretization::forms::force< TrialT, TestT >:



Collaboration diagram for serif::discretization::forms::force< TrialT, TestT >:



Public Types

- using trial_type = TrialT
  The trial term type for the force form.
- using test_type = TestT
  The test term type for the force form.
- using operand_types = utils::types::TypeList<Operand<TrialT>, Operand<TestT>>
  Type list of operand types for the force form, used for compile-time reflection and iteration over the operands.

Static Public Attributes

- static constexpr std::string_view name = meta::template_name<force>()
  Compile time reflected name of the form.

10.203.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::force< TrialT, TestT >

The force form represents the force term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the force operation.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

10.203.2 Member Typedef Documentation

10.203.2.1 operand_types

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::force< TrialT, TestT >::operand_types = utils::types::TypeList<Operand<TrialT>, Operand<TestT>>

Type list of operand types for the force form, used for compile-time reflection and iteration over the operands.

10.203.2.2 test_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::force< TrialT, TestT >::test_type = TestT

The test term type for the force form.

10.203.2.3 trial_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::force< TrialT, TestT >::trial_type = TrialT

The trial term type for the force form.

10.203.3 Member Data Documentation

10.203.3.1 name

template<blocks::Term TrialT, blocks::Term TestT>
std::string_view serif::discretization::forms::force< TrialT, TestT >::name = meta::template_name<force>() [static], [constexpr]

Compile time reflected name of the form.

The documentation for this struct was generated from the following file:

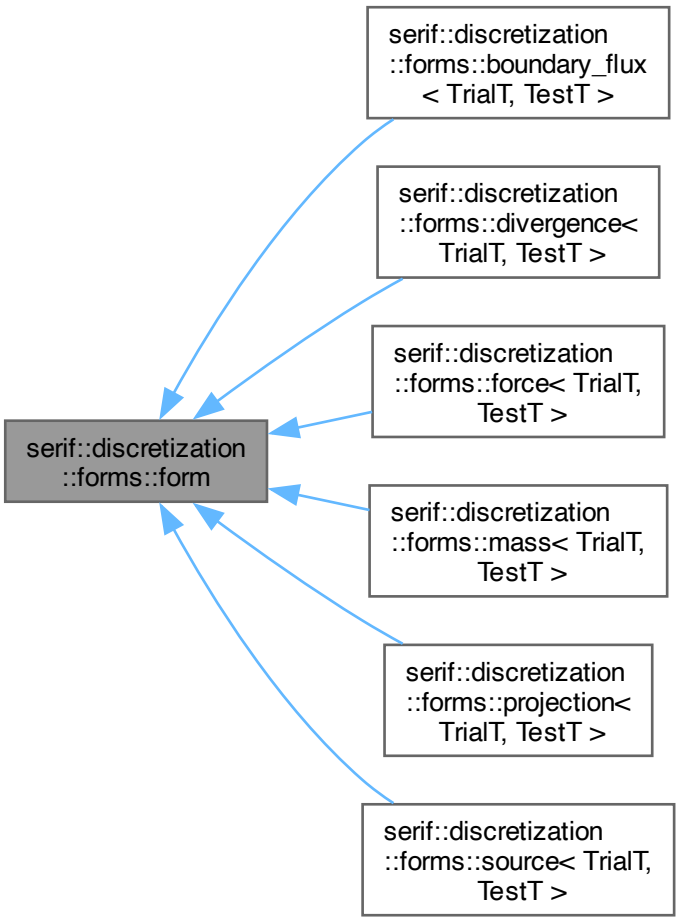
- src/include/serif/discretization/forms/forms.hpp

10.204 serif::discretization::forms::form Struct Reference

The form struct is a tag type that is used to identify a form in the discretization. It is used to group together forms that belong to the same mathematical operation. Each form should have a unique name and a list of operand types that belong to it.

#include <forms.hpp>

Inheritance diagram for serif::discretization::forms::form:



10.204.1 Detailed Description

The form struct is a tag type that is used to identify a form in the discretization. It is used to group together forms that belong to the same mathematical operation. Each form should have a unique name and a list of operand types that belong to it.

The documentation for this struct was generated from the following file:

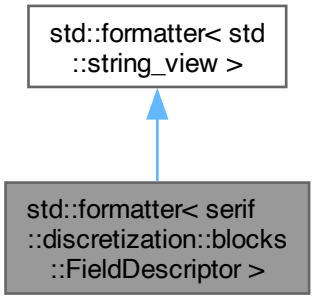
- src/include/serif/discretization/forms/forms.hpp

10.205 std::formatter< serif::discretization::blocks::FieldDescriptor > Struct Reference

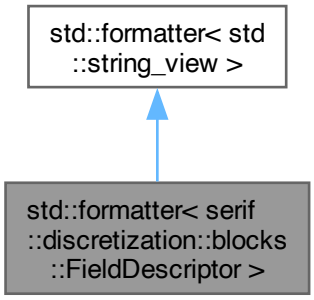
Specialization of std::formatter for FieldDescriptor and TermDescriptor to enable formatting with std::format.

#include <runtime.hpp>

Inheritance diagram for std::formatter< serif::discretization::blocks::FieldDescriptor >:



Collaboration diagram for std::formatter< serif::discretization::blocks::FieldDescriptor >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::blocks::FieldDescriptor &fd, FormatContext &ctx) const

10.205.1 Detailed Description

Specialization of std::formatter for FieldDescriptor and TermDescriptor to enable formatting with std::format.

10.205.2 Member Function Documentation

10.205.2.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::blocks::FieldDescriptor >::format (
                const serif::discretization::blocks::FieldDescriptor & fd,
                FormatContext & ctx) const    [inline]
```

The documentation for this struct was generated from the following file:

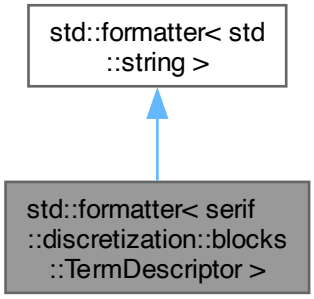
- src/include/serif/discretization/blocks/runtime.hpp

10.206 std::formatter< serif::discretization::blocks::TermDescriptor > Struct Reference

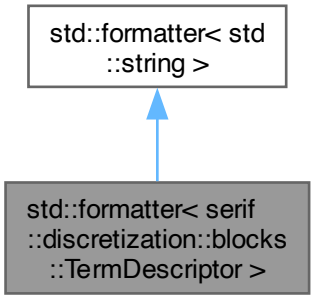
Specialization of std::formatter for TermDescriptor to enable formatting with std::format.

#include <runtime.hpp>

Inheritance diagram for std::formatter< serif::discretization::blocks::TermDescriptor >:



Collaboration diagram for std::formatter< serif::discretization::blocks::TermDescriptor >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::blocks::TermDescriptor &td, FormatContext &ctx) const

10.206.1 Detailed Description

Specialization of std::formatter for TermDescriptor to enable formatting with std::format.

10.206.2 Member Function Documentation

10.206.2.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::blocks::TermDescriptor >::format (
                const serif::discretization::blocks::TermDescriptor & td,
                FormatContext & ctx) const    [inline]
```

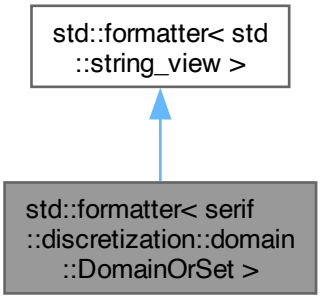
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/runtime.hpp

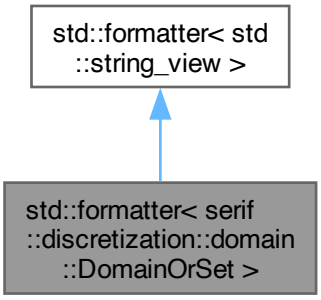
10.207 std::formatter< serif::discretization::domain::DomainOrSet > Struct Reference

#include <concepts.hpp>

Inheritance diagram for std::formatter< serif::discretization::domain::DomainOrSet >:



Collaboration diagram for std::formatter< serif::discretization::domain::DomainOrSet >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::domain::DomainOrSet &dos, FormatContext &ctx) const

10.207.1 Member Function Documentation

10.207.1.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::domain::DomainOrSet >::format (
                const serif::discretization::domain::DomainOrSet & dos,
                FormatContext & ctx) const   [inline]
```

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/concepts.hpp

10.208 std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult > Struct Reference

#include <results.hpp>

Public Member Functions

- constexpr auto parse (std::format_parse_context &ctx)
- auto format (const serif::discretization::domain::schema::validation::SchemaValidationResult &result, std::format_context &ctx) const

Public Attributes

- std::formatter< std::string > string_formatter

10.208.1 Member Function Documentation

10.208.1.1 format()

```
auto std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult >::format (
                const serif::discretization::domain::schema::validation::SchemaValidationResult & result,
                std::format_context & ctx) const   [inline]
```

10.208.1.2 parse()

```
auto std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult >::parse (
                std::format_parse_context & ctx)   [inline], [constexpr]
```

10.208.2 Member Data Documentation

10.208.2.1 string_formatter

std::formatter<std::string> std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult >↩
::string_formatter

The documentation for this struct was generated from the following file:

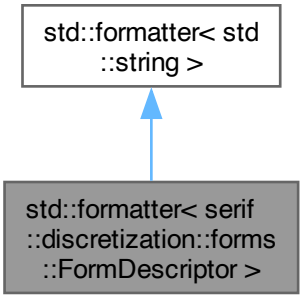
- src/include/serif/discretization/domain/schema/validation/results.hpp

10.209 std::formatter< serif::discretization::forms::FormDescriptor > Struct Reference

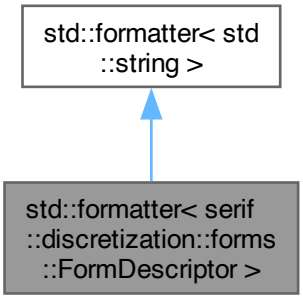
Overload to allow println for FormDescriptors.

#include <runtime.hpp>

Inheritance diagram for std::formatter< serif::discretization::forms::FormDescriptor >:



Collaboration diagram for std::formatter< serif::discretization::forms::FormDescriptor >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::forms::FormDescriptor &fd, FormatContext &ctx) const

10.209.1 Detailed Description

Overload to allow println for FormDescriptors.

10.209.2 Member Function Documentation

10.209.2.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::forms::FormDescriptor >::format (
                const serif::discretization::forms::FormDescriptor & fd,
                FormatContext & ctx) const    [inline]
```

The documentation for this struct was generated from the following file:

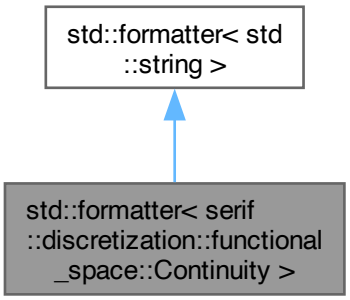
- src/include/serif/discretization/forms/runtime.hpp

10.210 std::formatter< serif::discretization::functional_space::Continuity > Struct Reference

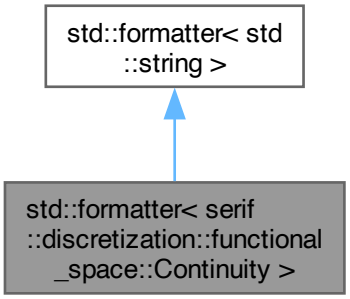
Specialization of the std::formatter for the Continuity enum. This allows us to use std::format to format the Continuity enum as strings.

#include <space.hpp>

Inheritance diagram for std::formatter< serif::discretization::functional_space::Continuity >:



Collaboration diagram for std::formatter< serif::discretization::functional_space::Continuity >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::functional_space::Continuity &continuity, FormatContext &ctx) const

10.210.1 Detailed Description

Specialization of the std::formatter for the Continuity enum. This allows us to use std::format to format the Continuity enum as strings.

10.210.2 Member Function Documentation

10.210.2.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::functional_space::Continuity >::format (
                const serif::discretization::functional_space::Continuity & continuity,
                FormatContext & ctx) const   [inline]
```

The documentation for this struct was generated from the following file:

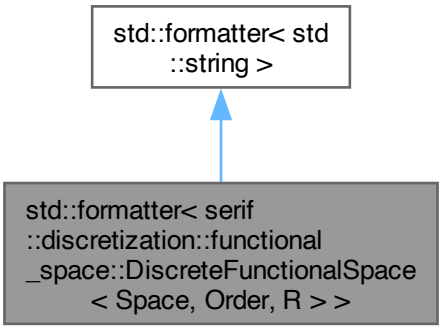
- src/include/serif/discretization/functional_space/space.hpp

10.211 std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > > Struct Template Reference

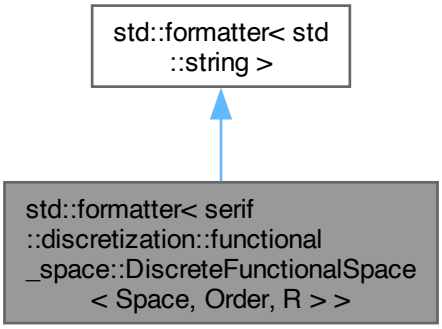
Specialization of the std::formatter for the DiscreteFunctionalSpace types. This allows us to use std::format to format the DiscreteFunctionalSpace types as strings.

#include <space.hpp>

Inheritance diagram for std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > >:



Collaboration diagram for std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > &space, FormatContext &ctx) const

10.211.1 Detailed Description

template<serif::discretization::functional_space::SpaceDescriptor Space, int Order, std::size_t R>
struct std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > >

Specialization of the std::formatter for the DiscreteFunctionalSpace types. This allows us to use std::format to format the DiscreteFunctionalSpace types as strings.

10.211.2 Member Function Documentation

10.211.2.1 format()

template<serif::discretization::functional_space::SpaceDescriptor Space, int Order, std::size_t R>
template<typename FormatContext>
auto std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > >::format (
const serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > & space,
FormatContext & ctx) const [inline]

The documentation for this struct was generated from the following file:

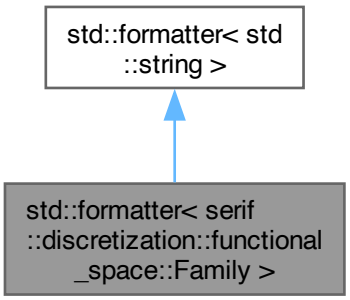
- src/include/serif/discretization/functional_space/space.hpp

10.212 std::formatter< serif::discretization::functional_space::Family > Struct Reference

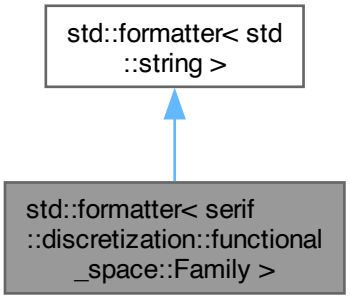
Specialization of the std::formatter for the Family enum. This allows us to use std::format to format the Family enum as strings.

#include <space.hpp>

Inheritance diagram for std::formatter< serif::discretization::functional_space::Family >:



Collaboration diagram for std::formatter< serif::discretization::functional_space::Family >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::functional_space::Family &family, FormatContext &ctx) const

10.212.1 Detailed Description

Specialization of the std::formatter for the Family enum. This allows us to use std::format to format the Family enum as strings.

10.212.2 Member Function Documentation

10.212.2.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::functional_space::Family >::format (
                const serif::discretization::functional_space::Family & family,
                FormatContext & ctx) const    [inline]
```

The documentation for this struct was generated from the following file:

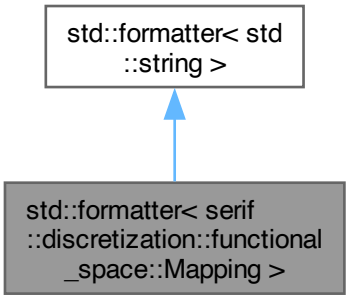
- src/include/serif/discretization/functional_space/space.hpp

10.213 std::formatter< serif::discretization::functional_space::Mapping > Struct Reference

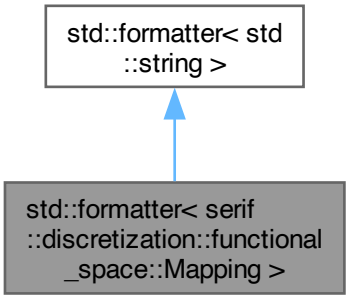
Specialization of the std::formatter for the Mapping enum. This allows us to use std::format to format the Mapping enum as strings.

#include <space.hpp>

Inheritance diagram for std::formatter< serif::discretization::functional_space::Mapping >:



Collaboration diagram for std::formatter< serif::discretization::functional_space::Mapping >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::functional_space::Mapping &mapping, FormatContext &ctx) const

10.213.1 Detailed Description

Specialization of the std::formatter for the Mapping enum. This allows us to use std::format to format the Mapping enum as strings.

10.213.2 Member Function Documentation

10.213.2.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::functional_space::Mapping >::format (
                const serif::discretization::functional_space::Mapping & mapping,
                FormatContext & ctx) const   [inline]
```

The documentation for this struct was generated from the following file:

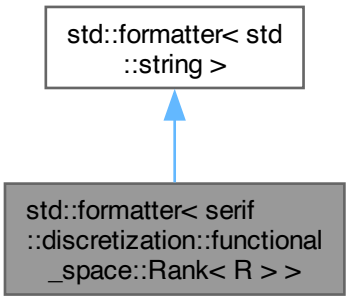
- src/include/serif/discretization/functional_space/space.hpp

10.214 std::formatter< serif::discretization::functional_space::Rank< R > > Struct Template Reference

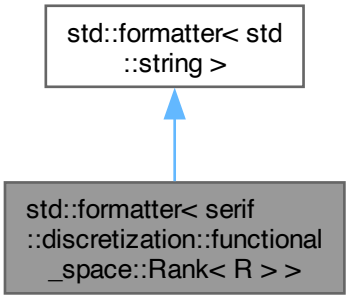
Specialization of the std::formatter for the functional space types. This allows us to use std::format to format the functional space types as strings.

#include <space.hpp>

Inheritance diagram for std::formatter< serif::discretization::functional_space::Rank< R > >:



Collaboration diagram for std::formatter< serif::discretization::functional_space::Rank< R > >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::functional_space::Rank< R > &, FormatContext &ctx) const

10.214.1 Detailed Description

template<std::size_t R>
struct std::formatter< serif::discretization::functional_space::Rank< R > >

Specialization of the std::formatter for the functional space types. This allows us to use std::format to format the functional space types as strings.

10.214.2 Member Function Documentation

10.214.2.1 format()

template<std::size_t R>
template<typename FormatContext>
auto std::formatter< serif::discretization::functional_space::Rank< R > >::format (
const serif::discretization::functional_space::Rank< R > & ,
FormatContext & ctx) const [inline]

The documentation for this struct was generated from the following file:

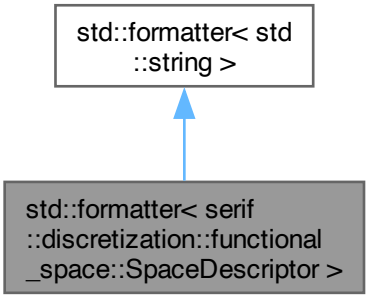
- src/include/serif/discretization/functional_space/space.hpp

10.215 std::formatter< serif::discretization::functional_space::SpaceDescriptor > Struct Reference

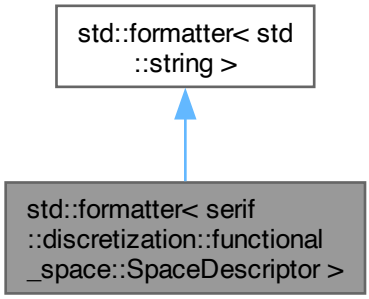
Specialization of the std::formatter for the SpaceDescriptor types. This allows us to use std::format to format the SpaceDescriptor types as strings.

#include <space.hpp>

Inheritance diagram for std::formatter< serif::discretization::functional_space::SpaceDescriptor >:



Collaboration diagram for std::formatter< serif::discretization::functional_space::SpaceDescriptor >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::functional_space::SpaceDescriptor &space, FormatContext &ctx) const

10.215.1 Detailed Description

Specialization of the std::formatter for the SpaceDescriptor types. This allows us to use std::format to format the SpaceDescriptor types as strings.

10.215.2 Member Function Documentation

10.215.2.1 format()

template<typename FormatContext>
auto std::formatter< serif::discretization::functional_space::SpaceDescriptor >::format (
const serif::discretization::functional_space::SpaceDescriptor & space,
FormatContext & ctx) const [inline]

The documentation for this struct was generated from the following file:

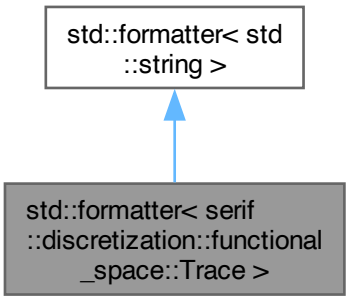
- src/include/serif/discretization/functional_space/space.hpp

10.216 std::formatter< serif::discretization::functional_space::Trace > Struct Reference

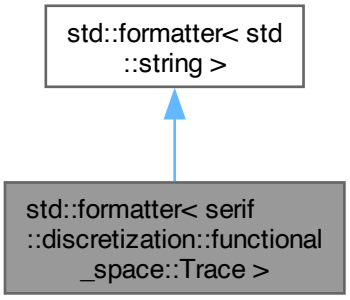
Specialization of the std::formatter for the Trace enum. This allows us to use std::format to format the Trace enum as strings.

#include <space.hpp>

Inheritance diagram for std::formatter< serif::discretization::functional_space::Trace >:



Collaboration diagram for std::formatter< serif::discretization::functional_space::Trace >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::functional_space::Trace &trace, FormatContext &ctx) const

10.216.1 Detailed Description

Specialization of the std::formatter for the Trace enum. This allows us to use std::format to format the Trace enum as strings.

10.216.2 Member Function Documentation

10.216.2.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::functional_space::Trace >::format (
                const serif::discretization::functional_space::Trace & trace,
                FormatContext & ctx) const    [inline]
```

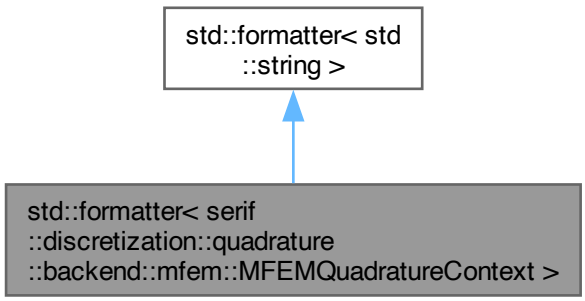
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/functional_space/space.hpp

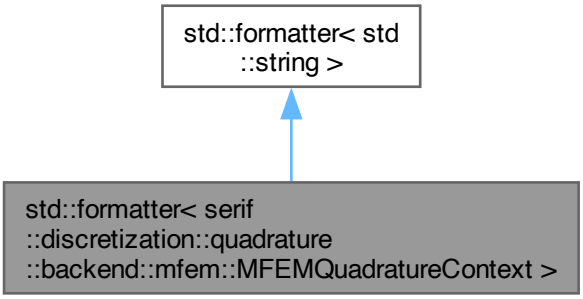
10.217 std::formatter< serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext > Struct Reference

#include <mfem_resolver.hpp>

Inheritance diagram for std::formatter< serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext >:



Collaboration diagram for std::formatter< serif::discretization::quadrature::backend::mfem::MFEMQuadrature↩ Context >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext &mfem_context, FormatContext &fc)

10.217.1 Member Function Documentation

10.217.1.1 format()

template<typename FormatContext>
auto std::formatter< serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext >::format (
const serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext & mfem_context,
FormatContext & fc) [inline]

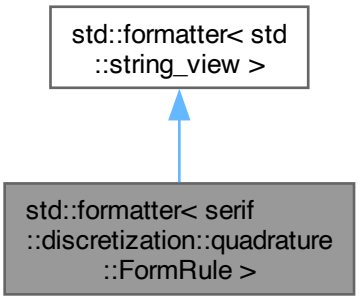
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/backend/mfem/mfem_resolver.hpp

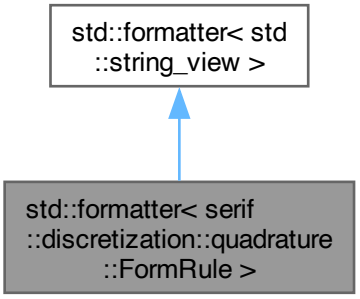
10.218 std::formatter< serif::discretization::quadrature::FormRule > Struct Reference

#include <rules.hpp>

Inheritance diagram for std::formatter< serif::discretization::quadrature::FormRule >:



Collaboration diagram for std::formatter< serif::discretization::quadrature::FormRule >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::quadrature::FormRule &fr, FormatContext &ctx) const

10.218.1 Member Function Documentation

10.218.1.1 format()

template<typename FormatContext>
auto std::formatter< serif::discretization::quadrature::FormRule >::format (

const serif::discretization::quadrature::FormRule & fr,
FormatContext & ctx) const [inline]

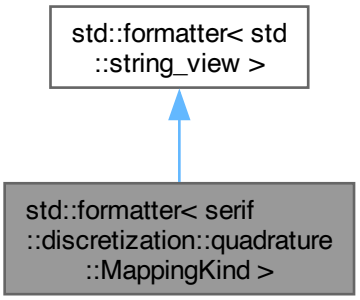
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

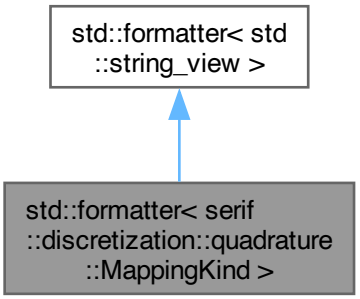
10.219 std::formatter< serif::discretization::quadrature::MappingKind > Struct Reference

#include <rules.hpp>

Inheritance diagram for std::formatter< serif::discretization::quadrature::MappingKind >:



Collaboration diagram for std::formatter< serif::discretization::quadrature::MappingKind >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::quadrature::MappingKind &mk, FormatContext &ctx) const

10.219.1 Member Function Documentation

10.219.1.1 format()

template<typename FormatContext>
auto std::formatter< serif::discretization::quadrature::MappingKind >::format (
const serif::discretization::quadrature::MappingKind & mk,
FormatContext & ctx) const [inline]

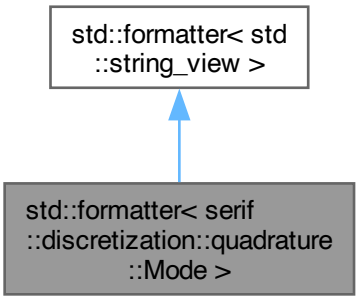
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

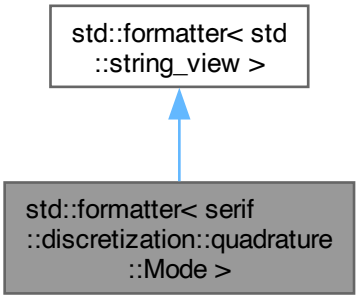
10.220 std::formatter< serif::discretization::quadrature::Mode > Struct Reference

#include <rules.hpp>

Inheritance diagram for std::formatter< serif::discretization::quadrature::Mode >:



Collaboration diagram for std::formatter< serif::discretization::quadrature::Mode >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::quadrature::Mode &mode, FormatContext &ctx) const

10.220.1 Member Function Documentation

10.220.1.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::quadrature::Mode >::format (
                const serif::discretization::quadrature::Mode & mode,
                FormatContext & ctx) const   [inline]
```

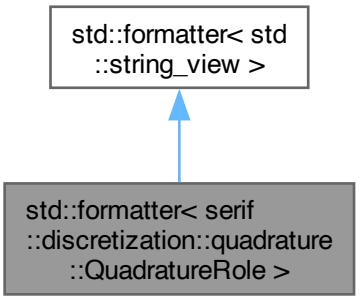
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

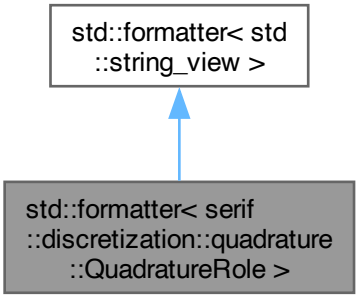
10.221 std::formatter< serif::discretization::quadrature::QuadratureRole > Struct Reference

#include <rules.hpp>

Inheritance diagram for std::formatter< serif::discretization::quadrature::QuadratureRole >:



Collaboration diagram for std::formatter< serif::discretization::quadrature::QuadratureRole >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::quadrature::QuadratureRole &role, FormatContext &ctx) const

10.221.1 Member Function Documentation

10.221.1.1 format()

template<typename FormatContext>
auto std::formatter< serif::discretization::quadrature::QuadratureRole >::format (

const serif::discretization::quadrature::QuadratureRole & role,
FormatContext & ctx) const [inline]

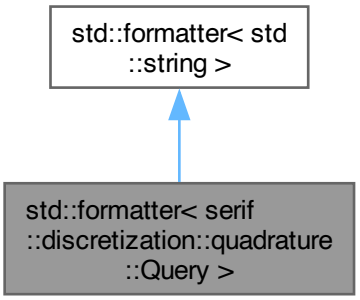
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

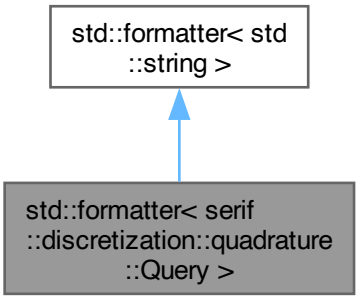
10.222 std::formatter< serif::discretization::quadrature::Query > Struct Reference

#include <query.hpp>

Inheritance diagram for std::formatter< serif::discretization::quadrature::Query >:



Collaboration diagram for std::formatter< serif::discretization::quadrature::Query >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::quadrature::Query &q, FormatContext &ctx) const

10.222.1 Member Function Documentation

10.222.1.1 format()

template<typename FormatContext>
auto std::formatter< serif::discretization::quadrature::Query >::format (
const serif::discretization::quadrature::Query & q,
FormatContext & ctx) const [inline]

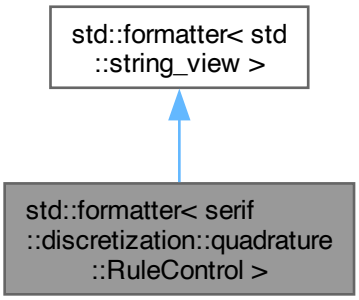
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/query.hpp

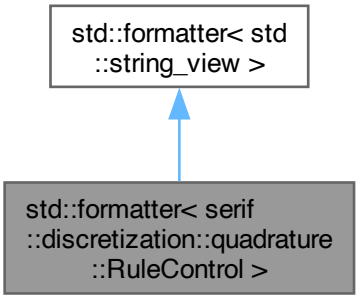
10.223 std::formatter< serif::discretization::quadrature::RuleControl > Struct Reference

#include <rules.hpp>

Inheritance diagram for std::formatter< serif::discretization::quadrature::RuleControl >:



Collaboration diagram for std::formatter< serif::discretization::quadrature::RuleControl >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::quadrature::RuleControl &rc, FormatContext &ctx) const

10.223.1 Member Function Documentation

10.223.1.1 format()

```
template<typename FormatContext>
auto std::formatter< serif::discretization::quadrature::RuleControl >::format (
                const serif::discretization::quadrature::RuleControl & rc,
                FormatContext & ctx) const   [inline]
```

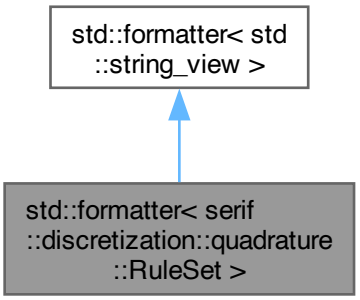
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

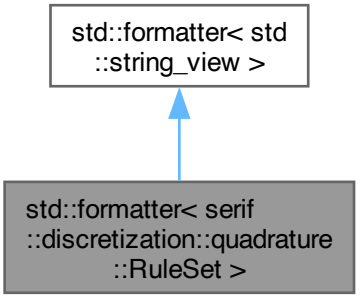
10.224 std::formatter< serif::discretization::quadrature::RuleSet > Struct Reference

#include <rules.hpp>

Inheritance diagram for std::formatter< serif::discretization::quadrature::RuleSet >:



Collaboration diagram for std::formatter< serif::discretization::quadrature::RuleSet >:



Public Member Functions

- template<typename FormatContext>
  auto format (const serif::discretization::quadrature::RuleSet &rs, FormatContext &ctx) const

10.224.1 Member Function Documentation

10.224.1.1 format()

template<typename FormatContext>
auto std::formatter< serif::discretization::quadrature::RuleSet >::format (

const serif::discretization::quadrature::RuleSet & rs,
FormatContext & ctx) const [inline]

The documentation for this struct was generated from the following file:

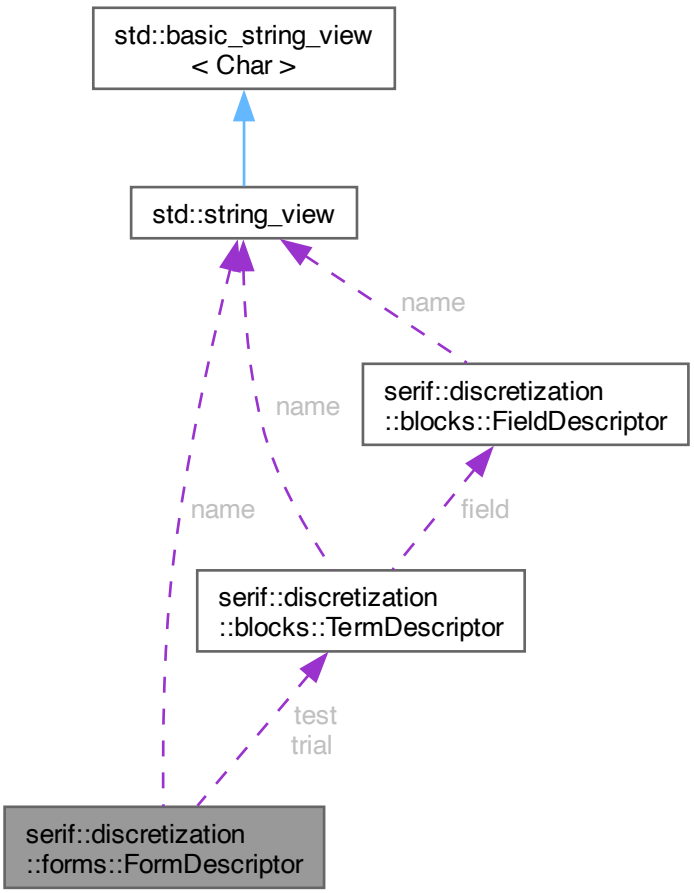
- src/include/serif/discretization/quadrature/rules.hpp

10.225 serif::discretization::forms::FormDescriptor Struct Reference

Runtime form descriptor including runtime term information. Used to represented the compile-time form type at runtime.

#include <runtime.hpp>

Collaboration diagram for serif::discretization::forms::FormDescriptor:



Public Attributes

- std::string_view name

  Name of the form usable at runtime.
- blocks::TermDescriptor trial

  Term descriptor for the trial term.
- blocks::TermDescriptor test

  Term descriptor for the test term.

Friends

- constexpr bool operator== (const FormDescriptor &, const FormDescriptor &) noexcept=default

  The standard struct equality operator. Two form descriptors are considered equal if and only if all members are considered equal. This means that the names are the same, the trial terms are the same, and the test terms are the same.

10.225.1 Detailed Description

Runtime form descriptor including runtime term information. Used to represented the compile-time form type at runtime.

10.225.2 Friends And Related Symbol Documentation

10.225.2.1 operator==

```
bool operator== (
            const FormDescriptor & ,
            const FormDescriptor & )   [friend]
```

The standard struct equality operator. Two form descriptors are considered equal if and only if all members are considered equal. This means that the names are the same, the trial terms are the same, and the test terms are the same.

Returns

10.225.3 Member Data Documentation

10.225.3.1 name

std::string_view serif::discretization::forms::FormDescriptor::name

Name of the form usable at runtime.

10.225.3.2 test

blocks::TermDescriptor serif::discretization::forms::FormDescriptor::test

Term descriptor for the test term.

10.225.3.3 trial

blocks::TermDescriptor serif::discretization::forms::FormDescriptor::trial

Term descriptor for the trial term.

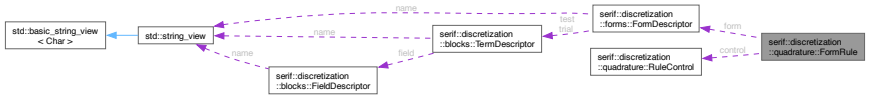
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/runtime.hpp

10.226 serif::discretization::quadrature::FormRule Struct Reference

#include <rules.hpp>

Collaboration diagram for serif::discretization::quadrature::FormRule:



Public Attributes

- forms::FormDescriptor form
- RuleControl control

10.226.1 Member Data Documentation

10.226.1.1 control

RuleControl serif::discretization::quadrature::FormRule::control

10.226.1.2 form

forms::FormDescriptor serif::discretization::quadrature::FormRule::form

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

10.227 serif::discretization::forms::FormTraits< FormT > Struct Template Reference

Compile-time traits for forms, used to extract types and properties from template parameters. This struct is specialized for each form type to provide specific traits such as trial and test order reductions.

#include <form_traits.hpp>

Static Public Attributes

- static constexpr int trial_order_reduction = 0
- static constexpr int test_order_reduction = 0

10.227.1 Detailed Description

template<Form FormT>
struct serif::discretization::forms::FormTraits< FormT >

Compile-time traits for forms, used to extract types and properties from template parameters. This struct is specialized for each form type to provide specific traits such as trial and test order reductions.

Template Parameters

FormT	The form type for which to extract traits.

10.227.2 Member Data Documentation

10.227.2.1 test_order_reduction

template<Form FormT>
int serif::discretization::forms::FormTraits< FormT >::test_order_reduction = 0 [static], [constexpr]

10.227.2.2 trial_order_reduction

template<Form FormT>
int serif::discretization::forms::FormTraits< FormT >::trial_order_reduction = 0 [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/form_traits.hpp

10.228 serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > > Struct Template Reference

Specialization of FormTraits for the boundary_flux form, which has no order reduction for either trial or test terms.

#include <form_traits.hpp>

Static Public Attributes

- static constexpr int trial_order_reduction = 0
- static constexpr int test_order_reduction = 0
- static constexpr int trial_order_reduction
- static constexpr int test_order_reduction

10.228.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > >

Specialization of FormTraits for the boundary_flux form, which has no order reduction for either trial or test terms.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

10.228.2 Member Data Documentation

10.228.2.1 test_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::test_order_reduction [static], [constexpr]

10.228.2.2 test_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > >::test_order_reduction = 0 [static], [constexpr]

10.228.2.3 trial_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::trial_order_reduction [static], [constexpr]

10.228.2.4 trial_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > >::trial_order_reduction = 0 [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/form_traits.hpp

10.229 serif::discretization::forms::FormTraits< divergence< TrialT, TestT > > Struct Template Reference

Specialization of FormTraits for the divergence form, which has an order reduction of 1, to satisfy the Inf-Sup condition, for the trial term and no order reduction for the test term.

#include <form_traits.hpp>

Static Public Attributes

- static constexpr int trial_order_reduction = 1
- static constexpr int test_order_reduction = 0
- static constexpr int trial_order_reduction
- static constexpr int test_order_reduction

10.229.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::FormTraits< divergence< TrialT, TestT > >

Specialization of FormTraits for the divergence form, which has an order reduction of 1, to satisfy the Inf-Sup condition, for the trial term and no order reduction for the test term.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

10.229.2 Member Data Documentation

10.229.2.1 test_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::test_order_reduction [static], [constexpr]

10.229.2.2 test_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< divergence< TrialT, TestT > >::test_order_reduction = 0 [static], [constexpr]

10.229.2.3 trial_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::trial_order_reduction [static], [constexpr]

10.229.2.4 trial_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< divergence< TrialT, TestT > >::trial_order_reduction = 1 [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/form_traits.hpp

10.230 serif::discretization::forms::FormTraits< force< TrialT, TestT > > Struct Template Reference

Specialization of FormTraits for the force form, which has no order reduction for either trial or test terms.

#include <form_traits.hpp>

Static Public Attributes

- static constexpr int trial_order_reduction = 0
- static constexpr int test_order_reduction = 0
- static constexpr int trial_order_reduction
- static constexpr int test_order_reduction

10.230.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::FormTraits< force< TrialT, TestT > >

Specialization of FormTraits for the force form, which has no order reduction for either trial or test terms.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

10.230.2 Member Data Documentation

10.230.2.1 test_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::test_order_reduction [static], [constexpr]

10.230.2.2 test_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< force< TrialT, TestT > >::test_order_reduction = 0 [static], [constexpr]

10.230.2.3 trial_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::trial_order_reduction [static], [constexpr]

10.230.2.4 trial_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< force< TrialT, TestT > >::trial_order_reduction = 0 [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/form_traits.hpp

10.231 serif::discretization::forms::FormTraits< mass< TrialT, TestT > > Struct Template Reference

Specialization of FormTraits for the mass form, which has no order reduction for either trial or test terms.

#include <form_traits.hpp>

Static Public Attributes

- static constexpr int trial_order_reduction = 0
- static constexpr int test_order_reduction = 0
- static constexpr int trial_order_reduction
- static constexpr int test_order_reduction

10.231.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::FormTraits< mass< TrialT, TestT > >

Specialization of FormTraits for the mass form, which has no order reduction for either trial or test terms.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

10.231.2 Member Data Documentation

10.231.2.1 test_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::test_order_reduction [static], [constexpr]

10.231.2.2 test_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< mass< TrialT, TestT > >::test_order_reduction = 0 [static], [constexpr]

10.231.2.3 trial_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::trial_order_reduction [static], [constexpr]

10.231.2.4 trial_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< mass< TrialT, TestT > >::trial_order_reduction = 0 [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/form_traits.hpp

10.232 serif::discretization::forms::FormTraits< projection< TrialT, TestT > > Struct Template Reference

Specialization of FormTraits for the projection form, which has no order reduction for either trial or test terms.

#include <form_traits.hpp>

Static Public Attributes

- static constexpr int trial_order_reduction = 0
- static constexpr int test_order_reduction = 0
- static constexpr int trial_order_reduction
- static constexpr int test_order_reduction

10.232.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::FormTraits< projection< TrialT, TestT > >

Specialization of FormTraits for the projection form, which has no order reduction for either trial or test terms.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

10.232.2 Member Data Documentation

10.232.2.1 test_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::test_order_reduction [static], [constexpr]

10.232.2.2 test_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< projection< TrialT, TestT > >::test_order_reduction = 0 [static], [constexpr]

10.232.2.3 trial_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::trial_order_reduction [static], [constexpr]

10.232.2.4 trial_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< projection< TrialT, TestT > >::trial_order_reduction = 0 [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/form_traits.hpp

10.233 serif::discretization::forms::FormTraits< source< TrialT, TestT > > Struct Template Reference

Specialization of FormTraits for the source form, which has no order reduction for either trial or test terms.

#include <form_traits.hpp>

Static Public Attributes

- static constexpr int trial_order_reduction = 0
- static constexpr int test_order_reduction = 0
- static constexpr int trial_order_reduction
- static constexpr int test_order_reduction

10.233.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::FormTraits< source< TrialT, TestT > >

Specialization of FormTraits for the source form, which has no order reduction for either trial or test terms.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

10.233.2 Member Data Documentation

10.233.2.1 test_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::test_order_reduction [static], [constexpr]

10.233.2.2 test_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< source< TrialT, TestT > >::test_order_reduction = 0 [static], [constexpr]

10.233.2.3 trial_order_reduction [1/2]

int serif::discretization::forms::FormTraits< FormT >::trial_order_reduction [static], [constexpr]

10.233.2.4 trial_order_reduction [2/2]

template<blocks::Term TrialT, blocks::Term TestT>
int serif::discretization::forms::FormTraits< source< TrialT, TestT > >::trial_order_reduction = 0 [static], [constexpr]

The documentation for this struct was generated from the following file:

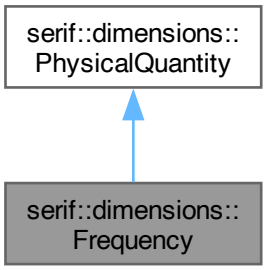
- src/include/serif/discretization/forms/form_traits.hpp

10.234 serif::dimensions::Frequency Struct Reference

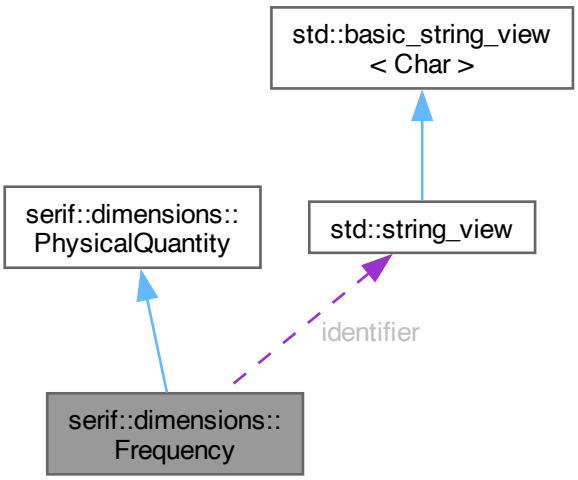
Dimension of Frequency.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Frequency:



Collaboration diagram for serif::dimensions::Frequency:



Static Public Attributes

- static constexpr std::string_view identifier = "frequency"

10.234.1 Detailed Description

Dimension of Frequency.

10.234.2 Member Data Documentation

10.234.2.1 identifier

std::string_view serif::dimensions::Frequency::identifier = ”frequency” [static], [constexpr]

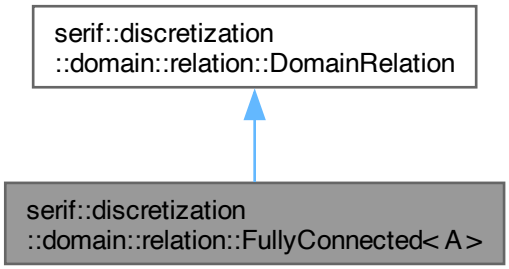
The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

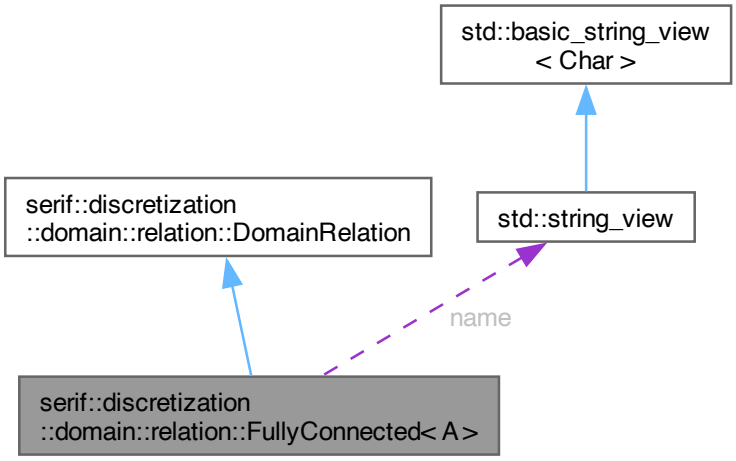
10.235 serif::discretization::domain::relation::FullyConnected< A > Struct Template Reference

#include <relations.hpp>

Inheritance diagram for serif::discretization::domain::relation::FullyConnected< A >:



Collaboration diagram for serif::discretization::domain::relation::FullyConnected< A >:



Public Types

- using domain_type = A

Static Public Attributes

- static constexpr std::string_view name = "fully_connected"

10.235.1 Member Typedef Documentation

10.235.1.1 domain_type

template<IsDomainOrSet A>
using serif::discretization::domain::relation::FullyConnected< A >::domain_type = A

10.235.2 Member Data Documentation

10.235.2.1 name

template<IsDomainOrSet A>
std::string_view serif::discretization::domain::relation::FullyConnected< A >::name = "fully_connected" [static], [constexpr]

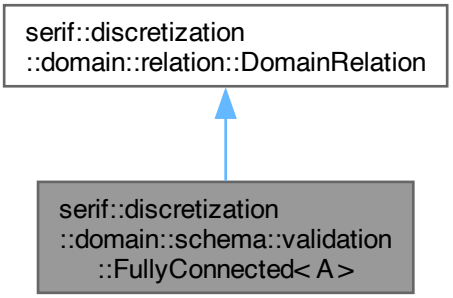
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/relations.hpp

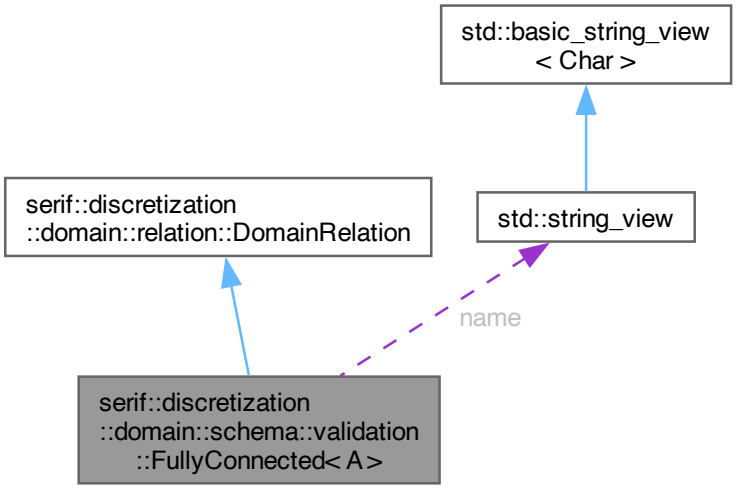
10.236 serif::discretization::domain::schema::validation::FullyConnected< A > Struct Template Reference

#include <relations.hpp>

Inheritance diagram for serif::discretization::domain::schema::validation::FullyConnected< A >:



Collaboration diagram for serif::discretization::domain::schema::validation::FullyConnected< A >:



Public Types

- using domain_type = A

Static Public Attributes

- static constexpr std::string_view name = "fully_connected"

10.236.1 Member Typedef Documentation

10.236.1.1 domain_type

template<IsDomainOrSet A>
using serif::discretization::domain::relation::FullyConnected< A >::domain_type = A

10.236.2 Member Data Documentation

10.236.2.1 name

template<IsDomainOrSet A>
std::string_view serif::discretization::domain::relation::FullyConnected< A >::name = "fully_connected" [static], [constexpr]

The documentation for this struct was generated from the following file:

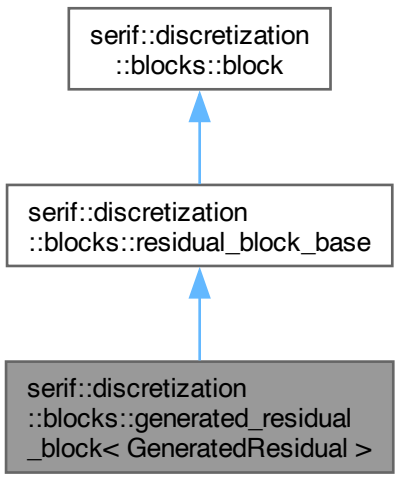
- src/include/serif/discretization/domain/relation/relations.hpp

10.237 serif::discretization::blocks::generated_residual_block< GeneratedResidual > Struct Template Reference

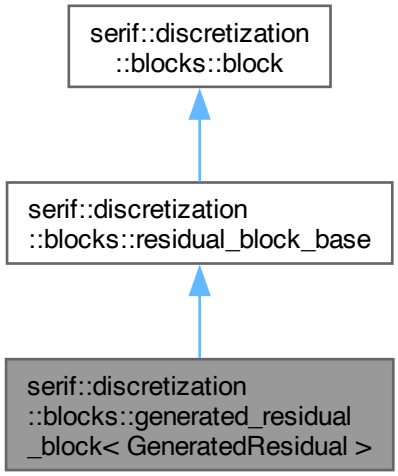
Final class for generated residual blocks, derived from residual_block_base. These are blocks that are generated at runtime.

#include <base.hpp>

Inheritance diagram for serif::discretization::blocks::generated_residual_block< GeneratedResidual >:



Collaboration diagram for serif::discretization::blocks::generated_residual_block< GeneratedResidual >:



Public Types

- using GeneratedType = GeneratedResidual

Static Public Attributes

- static constexpr int static_block_size = static_cast<int>(GeneratedResidual::scalarArity)

Static Public Attributes inherited from serif::discretization::blocks::residual_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.237.1 Detailed Description

template<typename GeneratedResidual>
struct serif::discretization::blocks::generated_residual_block< GeneratedResidual >

Final class for generated residual blocks, derived from residual_block_base. These are blocks that are generated at runtime.

Todo This may be removable at this point

10.237.2 Member Typedef Documentation

10.237.2.1 GeneratedType

template<typename GeneratedResidual>
using serif::discretization::blocks::generated_residual_block< GeneratedResidual >::GeneratedType = GeneratedResidual

10.237.3 Member Data Documentation

10.237.3.1 static_block_size

template<typename GeneratedResidual>
int serif::discretization::blocks::generated_residual_block< GeneratedResidual >::static_block_size = static_cast<int>(Generated↩
Residual::scalarArity) [static], [constexpr]

The documentation for this struct was generated from the following file:

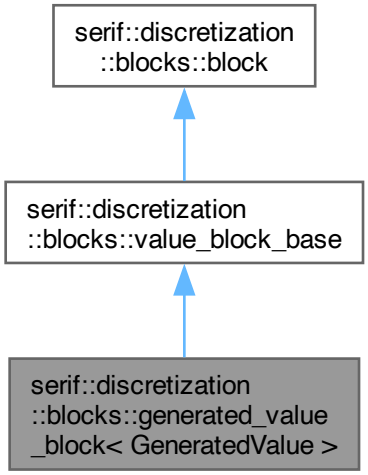
- src/include/serif/discretization/blocks/base.hpp

10.238 serif::discretization::blocks::generated_value_block< GeneratedValue > Struct Template Reference

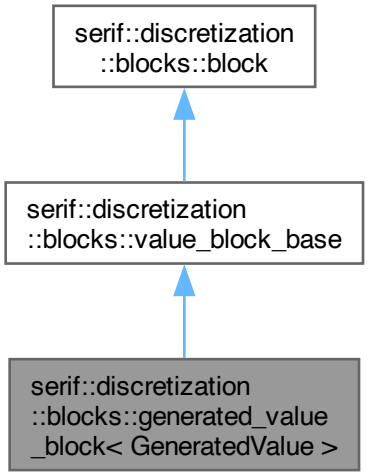
Final class for generated value blocks, derived from value_block_base. These are blocks that are generated at runtime.

#include <base.hpp>

Inheritance diagram for serif::discretization::blocks::generated_value_block< GeneratedValue >:



Collaboration diagram for serif::discretization::blocks::generated_value_block< GeneratedValue >:



Public Types

- using GeneratedType = GeneratedValue

Static Public Attributes

- static constexpr int static_block_size = static_cast<int>(GeneratedValue::scalarArity)

Static Public Attributes inherited from serif::discretization::blocks::value_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.238.1 Detailed Description

template<typename GeneratedValue>
struct serif::discretization::blocks::generated_value_block< GeneratedValue >

Final class for generated value blocks, derived from value_block_base. These are blocks that are generated at runtime.

Todo This may be removable at this point

10.238.2 Member Typedef Documentation

10.238.2.1 GeneratedType

template<typename GeneratedValue>
using serif::discretization::blocks::generated_value_block< GeneratedValue >::GeneratedType = GeneratedValue

10.238.3 Member Data Documentation

10.238.3.1 static_block_size

template<typename GeneratedValue>
int serif::discretization::blocks::generated_value_block< GeneratedValue >::static_block_size = static_cast<int>(Generated↩ Value::scalarArity) [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/base.hpp

10.239 serif::operators::backend::mfem::volume_keys::Geometry< Contribution > Struct Template Reference

#include <volume_data.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.240 serif::geometry::deformation::backend::mfem::GeometrySampleRule Struct Reference

Borrow one sampling rule; repeat an element to retain the actual union of downstream rules.

#include <geometry_step.hpp>

Public Attributes

- int element
- const ::mfem::IntegrationRule * integration_rule

10.240.1 Detailed Description

Borrow one sampling rule; repeat an element to retain the actual union of downstream rules.

10.240.2 Member Data Documentation

10.240.2.1 element

int serif::geometry::deformation::backend::mfem::GeometrySampleRule::element

10.240.2.2 integration_rule

const ::mfem::IntegrationRule* serif::geometry::deformation::backend::mfem::GeometrySampleRule::integration_rule

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/backend/mfem/geometry_step.hpp

10.241 serif::geometry::deformation::GeometryStepLimit Struct Reference

Geometry-only bound at the supplied sample points, with a location for the first obstruction.

#include <geometry_step.hpp>

Public Attributes

- double step {0.0}
- double boundary_step {0.0}
- double minimum_determinant_at_accepted {std::numeric_limits<double>::infinity()}
- double minimum_determinant_at_maximum {std::numeric_limits<double>::infinity()}
- double limiting_point_determinant_at_step {0.0}
- std::size_t sampled_point_count {0}
- bool limited_by_geometry {false}
- int limiting_element {-1}
- int limiting_rule {-1}
- int limiting_point {-1}

10.241.1 Detailed Description

Geometry-only bound at the supplied sample points, with a location for the first obstruction.

The determinant is that of the physical map relative to the reference mesh. This does not bound unsampled points, thermodynamic admissibility, or a nonlinear trajectory in shape coefficients. A boundary at maximum_step still applies fraction_to_boundary. An unrestricted step does not.

10.241.2 Member Data Documentation

10.241.2.1 boundary_step

double serif::geometry::deformation::GeometryStepLimit::boundary_step {0.0}

10.241.2.2 limited_by_geometry

bool serif::geometry::deformation::GeometryStepLimit::limited_by_geometry {false}

10.241.2.3 limiting_element

int serif::geometry::deformation::GeometryStepLimit::limiting_element {-1}

10.241.2.4 limiting_point

int serif::geometry::deformation::GeometryStepLimit::limiting_point {-1}

10.241.2.5 limiting_point_determinant_at_step

double serif::geometry::deformation::GeometryStepLimit::limiting_point_determinant_at_step {0.0}

10.241.2.6 limiting_rule

int serif::geometry::deformation::GeometryStepLimit::limiting_rule {-1}

10.241.2.7 minimum_determinant_at_accepted

double serif::geometry::deformation::GeometryStepLimit::minimum_determinant_at_accepted {std::numeric_limits<double>↩
::infinity()}

10.241.2.8 minimum_determinant_at_maximum

double serif::geometry::deformation::GeometryStepLimit::minimum_determinant_at_maximum {std::numeric_limits<double>↩
::infinity()}

10.241.2.9 sampled_point_count

std::size_t serif::geometry::deformation::GeometryStepLimit::sampled_point_count {0}

10.241.2.10 step

double serif::geometry::deformation::GeometryStepLimit::step {0.0}

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/geometry_step.hpp

10.242 serif::geometry::deformation::GeometryStepOptions Struct Reference

Numerical controls for keeping a straight volume-displacement step above a Jacobian determinant floor.

#include <geometry_step.hpp>

Public Attributes

- double maximum_step {1.0}
- double determinant_floor {0.0}
- double fraction_to_boundary {0.9}

10.242.1 Detailed Description

Numerical controls for keeping a straight volume-displacement step above a Jacobian determinant floor.

10.242.2 Member Data Documentation

10.242.2.1 determinant_floor

double serif::geometry::deformation::GeometryStepOptions::determinant_floor {0.0}

10.242.2.2 fraction_to_boundary

double serif::geometry::deformation::GeometryStepOptions::fraction_to_boundary {0.9}

10.242.2.3 maximum_step

double serif::geometry::deformation::GeometryStepOptions::maximum_step {1.0}

The documentation for this struct was generated from the following file:

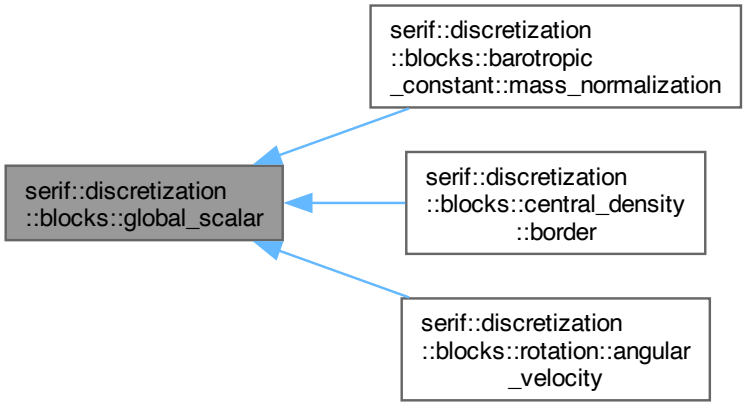
- src/include/serif/geometry/deformation/geometry_step.hpp

10.243 serif::discretization::blocks::global_scalar Struct Reference

A global scalar is a term that has a single value over the entire domain (think angular momentum or total mass). Broadly we need to separate these from the other terms since they require different (often less enriched) information at compile time.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::global_scalar:



Classes

- struct residual

  The residual block is a block that contains the residual of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the residual of the mass of the system (i.e. the difference between the current mass and the target mass).
- struct value

  The value block is a block that contains the value of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the actual value of the mass of the system.

10.243.1 Detailed Description

A global scalar is a term that has a single value over the entire domain (think angular momentum or total mass). Broadly we need to separate these from the other terms since they require different (often less enriched) information at compile time.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

10.244 serif::discretization::forms::Operation::Gradient Struct Reference

Operation tag for evaluating a gradient.

#include <operands.hpp>

10.244.1 Detailed Description

Operation tag for evaluating a gradient.

The documentation for this struct was generated from the following file:

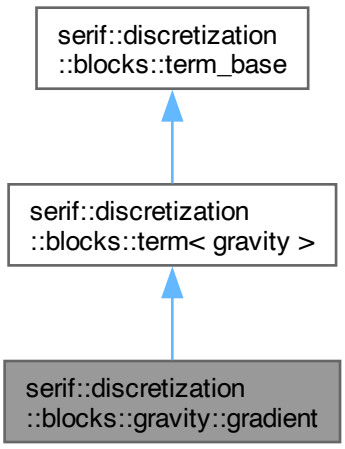
- src/include/serif/discretization/forms/operands.hpp

10.245 serif::discretization::blocks::gravity::gradient Struct Reference

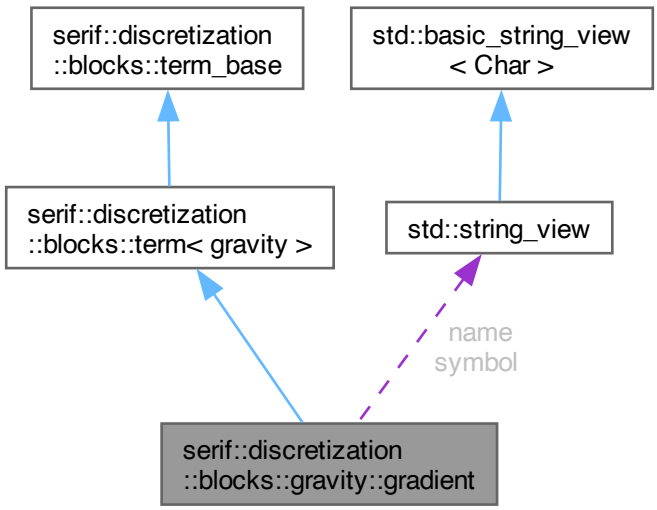
The gravitational gradient term is a 2nd order Hdiv vector field that is used to represent the negative gravitational acceleration of the system. Note that when I started developing this code I made the frankly wacky choice to let this be the true gradient of the potential, which is the negative acceleration. Broadly this may make sense mathematically; however, it means that developers must be extra careful to remember the sign convention.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::gravity::gradient:



Collaboration diagram for serif::discretization::blocks::gravity::gradient:



Classes

- struct residual

  The residual block is a block that contains the residual of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system.
- struct value

  The value block is a block that contains the value of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system.

Public Types

- using FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::Hdiv, 2, 1>

  2nd order Hdiv vector field. Read the template parameters as follows: 1) Hdiv space, 2) 2nd order, 3) rank 1 tensor field
- using Support = domain::AllDomains

  We support gravity on all domains, primarily since the only real boundary condition we have in the general case is that the potential goes to zero at infinity, which is in the vacuum domain.
- using Quantity = dimensions::Acceleration

  This is a dimension of acceleration, despite the wacky sign convention I adopted.

Public Types inherited from serif::discretization::blocks::term< gravity >

- using field_type

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<gradient>()

  Compile time reflected name of the term.
- static constexpr std::string_view symbol = " "

  Compile time symbol used for the term. Note that this is a UTF-8 character and may not render correctly in all environments. (TODO: Should we introduce an ASCII fallback?, something like make this a tuple of form <main, fallback>?)

10.245.1 Detailed Description

The gravitational gradient term is a 2nd order Hdiv vector field that is used to represent the negative gravitational acceleration of the system. Note that when I started developing this code I made the frankly wacky choice to let this be the true gradient of the potential, which is the negative acceleration. Broadly this may make sense mathematically; however, it means that developers must be extra careful to remember the sign convention.

10.245.2 Member Typedef Documentation

10.245.2.1 FunctionalSpace

using serif::discretization::blocks::gravity::gradient::FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::Hdiv, 2, 1>

2nd order Hdiv vector field. Read the template parameters as follows: 1) Hdiv space, 2) 2nd order, 3) rank 1 tensor field

10.245.2.2 Quantity

using serif::discretization::blocks::gravity::gradient::Quantity = dimensions::Acceleration

This is a dimension of acceleration, despite the wacky sign convention I adopted.

10.245.2.3 Support

using serif::discretization::blocks::gravity::gradient::Support = domain::AllDomains

We support gravity on all domains, primarily since the only real boundary condition we have in the general case is that the potential goes to zero at infinity, which is in the vacuum domain.

10.245.3 Member Data Documentation

10.245.3.1 name

std::string_view serif::discretization::blocks::gravity::gradient::name = meta::type_name<gradient>() [static], [constexpr]

Compile time reflected name of the term.

10.245.3.2 symbol

std::string_view serif::discretization::blocks::gravity::gradient::symbol = " " [static], [constexpr]

Compile time symbol used for the term. Note that this is a UTF-8 character and may not render correctly in all environments. (TODO: Should we introduce an ASCII fallback?, something like make this a tuple of form <main, fallback>?)

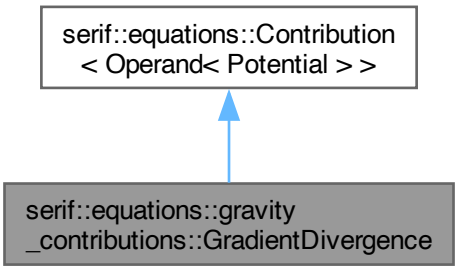
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

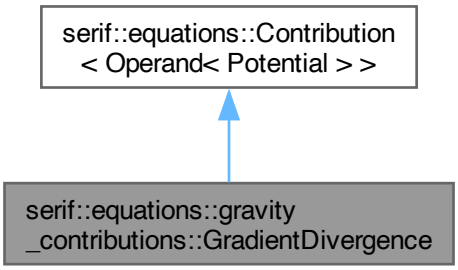
10.246 serif::equations::gravity_contributions::GradientDivergence Struct Reference

#include <mixed_gravity.hpp>

Inheritance diagram for serif::equations::gravity_contributions::GradientDivergence:



Collaboration diagram for serif::equations::gravity_contributions::GradientDivergence:



Public Types

- using Operands = utils::types::TypeList<Operand<Gradient, Divergence>>

Public Types inherited from serif::equations::Contribution< Operand< Potential > >

- using Test
- using TestOperand
- using Support

Public Member Functions

- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const
  Poisson's volume row contains + q div(g).
- int additional_order (Operand< Gradient, Divergence >, int) const

10.246.1 Member Typedef Documentation

10.246.1.1 Operands

using serif::equations::gravity_contributions::GradientDivergence::Operands = utils::types::TypeList<Operand<Gradient, Divergence>>

10.246.2 Member Function Documentation

10.246.2.1 additional_order()

int serif::equations::gravity_contributions::GradientDivergence::additional_order (
Operand< Gradient, Divergence > ,
int) const [inline]

10.246.2.2 evaluate()

PointLinearization< Operands > serif::equations::gravity_contributions::GradientDivergence::evaluate (
const EquationPoint< Operands > & point) const [inline]

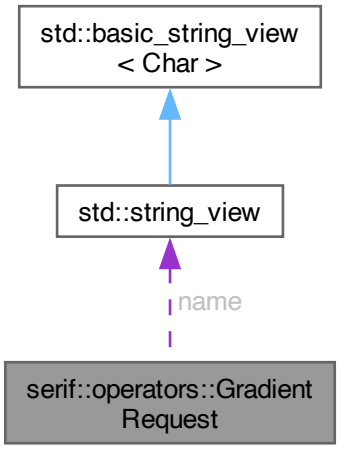
Poisson's volume row contains + q div(g).

The documentation for this struct was generated from the following file:

- src/include/serif/equations/mixed_gravity.hpp

10.247 serif::operators::GradientRequest Struct Reference

Collaboration diagram for serif::operators::GradientRequest:



Public Types

- using Test = Density
- using Operands = utils::types::TypeList<EnthalpyGradient>

Public Member Functions

- equations::PointLinearization< Operands > evaluate (const equations::EquationPoint< Operands > &) const

  Supply a syntactically valid authoring interface for the capability check.
- int additional_order (EnthalpyGradient, int) const

  Supply the existing numerical-order hook without enabling gradient evaluation.

Static Public Attributes

- static constexpr std::string_view name = "unsupported gradient request"

10.247.1 Member Typedef Documentation

10.247.1.1 Operands

using serif::operators::GradientRequest::Operands = utils::types::TypeList<EnthalpyGradient>

10.247.1.2 Test

using serif::operators::GradientRequest::Test = Density

10.247.2 Member Function Documentation

10.247.2.1 additional_order()

int serif::operators::GradientRequest::additional_order (
EnthalpyGradient ,
int) const [inline]

Supply the existing numerical-order hook without enabling gradient evaluation.

10.247.2.2 evaluate()

equations::PointLinearization< Operands > serif::operators::GradientRequest::evaluate (
const equations::EquationPoint< Operands > &) const [inline]

Supply a syntactically valid authoring interface for the capability check.

10.247.3 Member Data Documentation

10.247.3.1 name

std::string_view serif::operators::GradientRequest::name = "unsupported gradient request" [static], [constexpr]

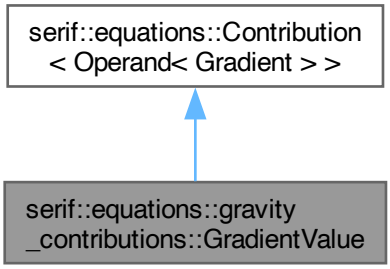
The documentation for this struct was generated from the following file:

- src/compile_time_checks/serif/operators/plan.cpp

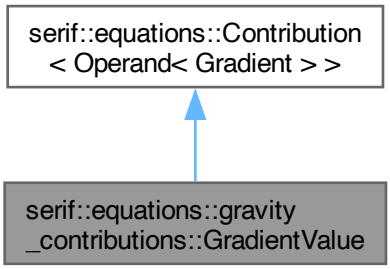
10.248 serif::equations::gravity_contributions::GradientValue Struct Reference

#include <mixed_gravity.hpp>

Inheritance diagram for serif::equations::gravity_contributions::GradientValue:



Collaboration diagram for serif::equations::gravity_contributions::GradientValue:



Public Types

- using Operands = utils::types::TypeList<Operand<Gradient>>

Public Types inherited from serif::equations::Contribution< Operand< Gradient > >

- using Test
- using TestOperand
- using Support

Public Member Functions

- PointLinearization< Operands, discretization::forms::VectorValue > evaluate (const EquationPoint< Operands > &point) const

  The coefficient of the vector test is g; its derivative with respect to g is I.
- int additional_order (Operand< Gradient >, int) const

10.248.1 Member Typedef Documentation

10.248.1.1 Operands

using serif::equations::gravity_contributions::GradientValue::Operands = utils::types::TypeList<Operand<Gradient>>

10.248.2 Member Function Documentation

10.248.2.1 additional_order()

int serif::equations::gravity_contributions::GradientValue::additional_order (
Operand< Gradient > ,
int) const [inline]

10.248.2.2 evaluate()

PointLinearization< Operands, discretization::forms::VectorValue > serif::equations::gravity_contributions::GradientValue::evaluate (
const EquationPoint< Operands > & point) const [inline]

The coefficient of the vector test is g; its derivative with respect to g is I.

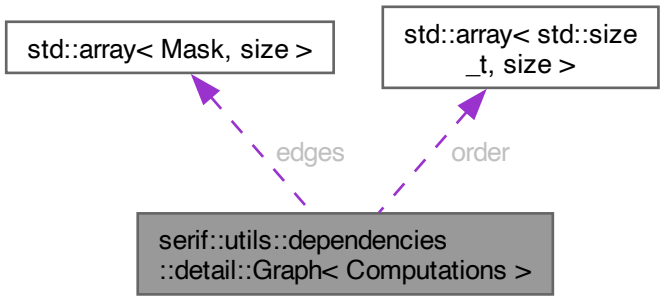
The documentation for this struct was generated from the following file:

- src/include/serif/equations/mixed_gravity.hpp

10.249 serif::utils::dependencies::detail::Graph< Computations > Struct Template Reference

#include <graph.hpp>

Collaboration diagram for serif::utils::dependencies::detail::Graph< Computations >:



Public Types

- using Mask = std::array<bool, size>
- using Matrix = std::array<Mask, size>
- using Order = std::array<std::size_t, size>

Static Public Member Functions

- template<typename ComputationT>
  static consteval Mask prerequisites ()

  Record only computed prerequisites; other input keys are supplied by a context.
- static consteval Order make_order ()

  Stable topological ordering makes shared prerequisites execute before their users.
- static consteval bool acyclic ()

  A missing position identifies a cycle, including a computation requiring itself.
- template<typename... Keys>
  static consteval Mask required (KeyList< Keys... >)

  Expand requested results to include all computed prerequisites.

Static Public Attributes

- static constexpr std::size_t size = sizeof...(Computations)
- static constexpr Matrix edges {prerequisites<Computations>()...}
- static constexpr Order order = make_order()

10.249.1 Member Typedef Documentation

10.249.1.1 Mask

template<typename... Computations>
using serif::utils::dependencies::detail::Graph< Computations >::Mask = std::array<bool, size>

10.249.1.2 Matrix

template<typename... Computations>
using serif::utils::dependencies::detail::Graph< Computations >::Matrix = std::array<Mask, size>

10.249.1.3 Order

template<typename... Computations>
using serif::utils::dependencies::detail::Graph< Computations >::Order = std::array<std::size_t, size>

10.249.2 Member Function Documentation

10.249.2.1 acyclic()

template<typename... Computations>
static consteval bool serif::utils::dependencies::detail::Graph< Computations >::acyclic () [inline], [static], [nodiscard], [consteval]

A missing position identifies a cycle, including a computation requiring itself.

10.249.2.2 make_order()

template<typename... Computations>
static consteval Order serif::utils::dependencies::detail::Graph< Computations >::make_order () [inline], [static], [nodiscard], [consteval]

Stable topological ordering makes shared prerequisites execute before their users.

10.249.2.3 prerequisites()

template<typename... Computations>
template<typename ComputationT>
static consteval Mask serif::utils::dependencies::detail::Graph< Computations >::prerequisites () [inline], [static], [nodiscard], [consteval]

Record only computed prerequisites; other input keys are supplied by a context.

10.249.2.4 required()

template<typename... Computations>
template<typename... Keys>
static consteval Mask serif::utils::dependencies::detail::Graph< Computations >::required (
KeyList< Keys... >) [inline], [static], [nodiscard], [consteval]

Expand requested results to include all computed prerequisites.

10.249.3 Member Data Documentation

10.249.3.1 edges

template<typename... Computations>
Matrix serif::utils::dependencies::detail::Graph< Computations >::edges {prerequisites<Computations>()...} [static], [constexpr]

10.249.3.2 order

template<typename... Computations>
Order serif::utils::dependencies::detail::Graph< Computations >::order = make_order() [static], [constexpr]

10.249.3.3 size

template<typename... Computations>
std::size_t serif::utils::dependencies::detail::Graph< Computations >::size = sizeof...(Computations) [static], [constexpr]

The documentation for this struct was generated from the following file:

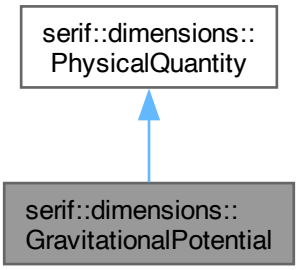
- src/include/serif/utils/dependencies/detail/graph.hpp

10.250 serif::dimensions::GravitationalPotential Struct Reference

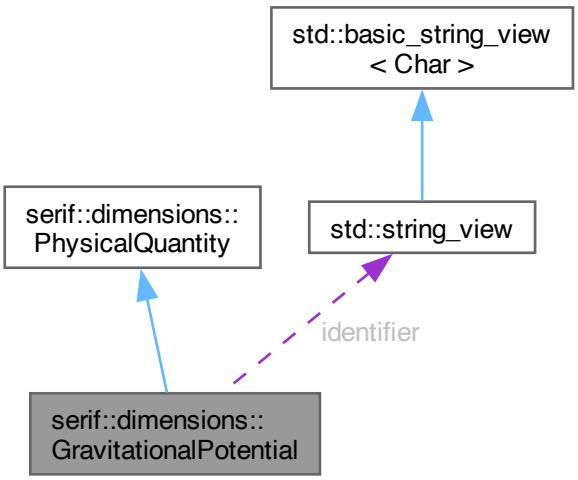
Dimension of Gravitational Potential.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::GravitationalPotential:



Collaboration diagram for serif::dimensions::GravitationalPotential:



Static Public Attributes

- static constexpr std::string_view identifier = "gravitational_potential"

10.250.1 Detailed Description

Dimension of Gravitational Potential.

10.250.2 Member Data Documentation

10.250.2.1 identifier

std::string_view serif::dimensions::GravitationalPotential::identifier = ”gravitational_potential” [static], [constexpr]

The documentation for this struct was generated from the following file:

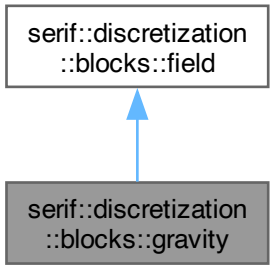
- src/include/serif/dimensions/dimensions.hpp

10.251 serif::discretization::blocks::gravity Struct Reference

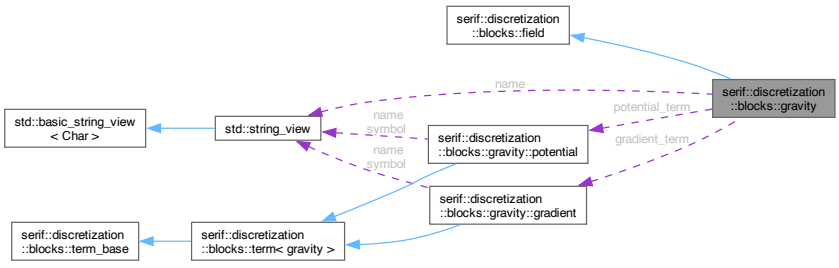
The gravity field is represented as a mixed system, in many ways this field is the motivation for the entire design of this file. Since it is a mixed system we have two separate terms that make it up, the potential and the gradient. These are not derived from one and other, rather they are weakly enforced to agree. This buys us an additional order of convergence in the potential which in tern allows us to use a lower resolution mesh.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::gravity:



Collaboration diagram for serif::discretization::blocks::gravity:



Classes

- struct gradient

  The gravitational gradient term is a 2nd order Hdiv vector field that is used to represent the negative gravitational acceleration of the system. Note that when I started developing this code I made the frankly wacky choice to let this be the true gradient of the potential, which is the negative acceleration. Broadly this may make sense mathematically; however, it means that developers must be extra careful to remember the sign convention.
- struct potential

  The gravitational potential term is a 2nd order L2 scalar field that is used to represent the gravitational potential of the system. Note that this is not exactly the same as the if you were to take the scalar potential of the gravitational gradient term. These are solved for by the code separately and weakly enforced to agree. Generally if you need the most accurate gradient you should use the gradient term and if you need the most accurate potential you should use this term (this is because differentiating this term loses an order of accuracy in the differentiation).

Public Types

- using Terms = utils::types::TypeList<potential, gradient>

  Type list of terms for compile time reflection and iteration over the terms of the field.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<gravity>()

  Compile time reflected name of the field.
- static constexpr potential potential_term {}

  Static constexpr instance of the potential term. This is used to access the term in the discretization.
- static constexpr gradient gradient_term {}

  Static constexpr instance of the gradient term. This is used to access the term in the discretization.

10.251.1 Detailed Description

The gravity field is represented as a mixed system, in many ways this field is the motivation for the entire design of this file. Since it is a mixed system we have two separate terms that make it up, the potential and the gradient. These are not derived from one and other, rather they are weakly enforced to agree. This buys us an additional order of convergence in the potential which in tern allows us to use a lower resolution mesh.

10.251.2 Member Typedef Documentation

10.251.2.1 Terms

using serif::discretization::blocks::gravity::Terms = utils::types::TypeList<potential, gradient>

Type list of terms for compile time reflection and iteration over the terms of the field.

10.251.3 Member Data Documentation

10.251.3.1 gradient_term

gradient serif::discretization::blocks::gravity::gradient_term {} [static], [constexpr]

Static constexpr instance of the gradient term. This is used to access the term in the discretization.

10.251.3.2 name

std::string_view serif::discretization::blocks::gravity::name = meta::type_name<gravity>() [static], [constexpr]

Compile time reflected name of the field.

10.251.3.3 potential_term

potential serif::discretization::blocks::gravity::potential_term {} [static], [constexpr]

Static constexpr instance of the potential term. This is used to access the term in the discretization.

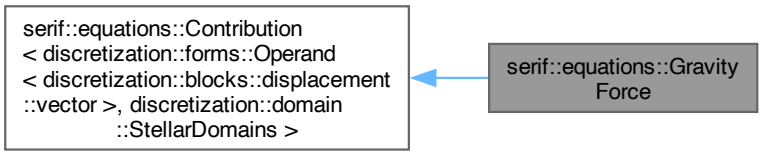
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

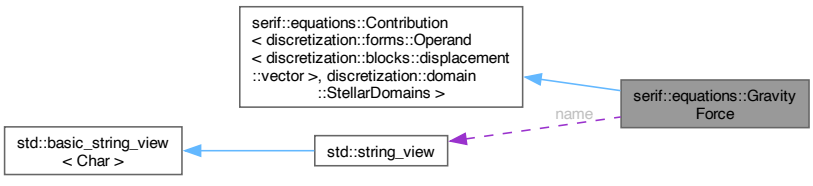
10.252 serif::equations::GravityForce Struct Reference

#include <gravity_force.hpp>

Inheritance diagram for serif::equations::GravityForce:



Collaboration diagram for serif::equations::GravityForce:



Public Types

- using Density = discretization::blocks::density::mass
- using Gradient = discretization::blocks::gravity::gradient
- using DensityValue = discretization::forms::Operand<Density>
- using GradientValue = discretization::forms::Operand<Gradient>
- using Operands = utils::types::TypeList<DensityValue, GradientValue>
- using Result = discretization::forms::VectorValue

Public Types inherited from serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::displacement::vecto

- using Test
- using TestOperand
- using Support

Public Member Functions

- PointLinearization< Operands, Result > evaluate (const EquationPoint< Operands > &point) const
  F = rho g, dF/drho = g, dF/dg = rho I.
- template<typename OperandT>
  int additional_order (OperandT, int) const

Static Public Attributes

- static constexpr std::string_view name = "gravitational force"

10.252.1 Detailed Description

Gravity's contribution to displacement balance: integral_star rho g . w dV. Here g = grad(Phi); for some reason which only past me can fathom I adopted g = grad(Phi) rather than g=-grad(Phi). The implication of this is that acceleration is -g. We may want to change this but we will need to be careful.

10.252.2 Member Typedef Documentation

10.252.2.1 Density

using serif::equations::GravityForce::Density = discretization::blocks::density::mass

10.252.2.2 DensityValue

using serif::equations::GravityForce::DensityValue = discretization::forms::Operand<Density>

10.252.2.3 Gradient

using serif::equations::GravityForce::Gradient = discretization::blocks::gravity::gradient

10.252.2.4 GradientValue

using serif::equations::GravityForce::GradientValue = discretization::forms::Operand<Gradient>

10.252.2.5 Operands

using serif::equations::GravityForce::Operands = utils::types::TypeList<DensityValue, GradientValue>

10.252.2.6 Result

using serif::equations::GravityForce::Result = discretization::forms::VectorValue

10.252.3 Member Function Documentation

10.252.3.1 additional_order()

template<typename OperandT>
int serif::equations::GravityForce::additional_order (
OperandT ,
int) const [inline], [nodiscard]

10.252.3.2 evaluate()

PointLinearization< Operands, Result > serif::equations::GravityForce::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

F = rho g, dF/drho = g, dF/dg = rho I.

10.252.4 Member Data Documentation

10.252.4.1 name

std::string_view serif::equations::GravityForce::name = "gravitational force" [static], [constexpr]

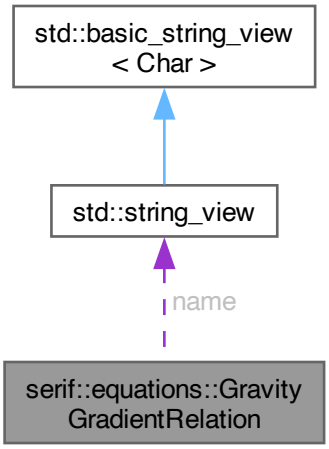
The documentation for this struct was generated from the following file:

- src/include/serif/equations/gravity_force.hpp

10.253 serif::equations::GravityGradientRelation Struct Reference

#include <mixed_gravity.hpp>

Collaboration diagram for serif::equations::GravityGradientRelation:



Public Types

- using Test = discretization::blocks::gravity::gradient

Public Member Functions

- auto contributions () const

Static Public Attributes

- static constexpr std::string_view name = "gravity-gradient relation"

10.253.1 Member Typedef Documentation

10.253.1.1 Test

using serif::equations::GravityGradientRelation::Test = discretization::blocks::gravity::gradient

10.253.2 Member Function Documentation

10.253.2.1 contributions()

auto serif::equations::GravityGradientRelation::contributions () const [inline]

10.253.3 Member Data Documentation

10.253.3.1 name

std::string_view serif::equations::GravityGradientRelation::name = "gravity-gradient relation" [static], [constexpr]

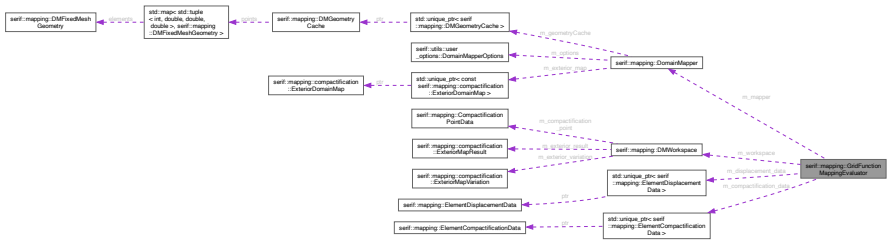
The documentation for this struct was generated from the following file:

- src/include/serif/equations/mixed_gravity.hpp

10.254 serif::mapping::GridFunctionMappingEvaluator Class Reference

#include <evaluator.hpp>

Collaboration diagram for serif::mapping::GridFunctionMappingEvaluator:



Public Member Functions

- GridFunctionMappingEvaluator (const DomainMapper &mapper, const mfem::GridFunction &displacement, const mfem::GridFunction &compactification_coordinate)
- void invalidate_cache () noexcept
- void refresh ()
- MappingStatus evaluate_point (mfem::ElementTransformation &transformation, const mfem::↩ IntegrationPoint &integration_point, MappingPointContext &context)
- MappingStatus evaluate_volume (mfem::ElementTransformation &transformation, const mfem::↩ IntegrationPoint &integration_point, VolumeMappingContext &context)
- MappingStatus evaluate_face (mfem::FaceElementTransformations &transformation, FaceElementSide side, const mfem::IntegrationPoint &integration_point, FaceMappingContext &context)
- VolumeQuadratureContext get_quadrature_context (mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &integration_point)
- FaceQuadratureContext get_face_quadrature_context (mfem::FaceElementTransformations &transformation, const mfem::IntegrationPoint &integration_point, FaceElementSide side=FaceElementSide::ELEMENT_1)
- void get_physical_point (mfem::ElementTransformation &transformation, const mfem::Integration↩ Point &integration_point, mfem::Vector &physical_position)

Private Member Functions

- void validate_field_bindings () const
- bool invalidate_for_changed_spaces ()
- void load_element (int element_id)

Private Attributes

- const DomainMapper & m_mapper
- const mfem::GridFunction & m_displacement
- const mfem::GridFunction & m_compactification_coordinate
- const mfem::FiniteElementSpace * m_displacement_space
- const mfem::FiniteElementSpace * m_compactification_space
- long m_displacement_space_sequence
- long m_compactification_space_sequence
- DMWorkspace m_workspace
- mfem::Array< int > m_displacement_dofs
- mfem::Array< int > m_compactification_dofs
- mfem::Vector m_element_displacement
- mfem::Vector m_element_compactification
- std::unique_ptr< ElementDisplacementData > m_displacement_data
- std::unique_ptr< ElementCompactificationData > m_compactification_data
- int m_cached_element_id {-1}

10.254.1 Constructor & Destructor Documentation

10.254.1.1 GridFunctionMappingEvaluator()

serif::mapping::GridFunctionMappingEvaluator::GridFunctionMappingEvaluator (
const DomainMapper & mapper,
const mfem::GridFunction & displacement,
const mfem::GridFunction & compactification_coordinate)

10.254.2 Member Function Documentation

10.254.2.1 evaluate_face()

MappingStatus serif::mapping::GridFunctionMappingEvaluator::evaluate_face (
mfem::FaceElementTransformations & transformation,
FaceElementSide side,
const mfem::IntegrationPoint & integration_point,
FaceMappingContext & context) [nodiscard]

10.254.2.2 evaluate_point()

MappingStatus serif::mapping::GridFunctionMappingEvaluator::evaluate_point (
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & integration_point,
MappingPointContext & context) [nodiscard]

10.254.2.3 evaluate_volume()

MappingStatus serif::mapping::GridFunctionMappingEvaluator::evaluate_volume (
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & integration_point,
VolumeMappingContext & context) [nodiscard]

10.254.2.4 get_face_quadrature_context()

FaceQuadratureContext serif::mapping::GridFunctionMappingEvaluator::get_face_quadrature_context (
mfem::FaceElementTransformations & transformation,
const mfem::IntegrationPoint & integration_point,
FaceElementSide side = FaceElementSide::ELEMENT_1) [nodiscard]

10.254.2.5 get_physical_point()

void serif::mapping::GridFunctionMappingEvaluator::get_physical_point (
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & integration_point,
mfem::Vector & physical_position)

10.254.2.6 get_quadrature_context()

VolumeQuadratureContext serif::mapping::GridFunctionMappingEvaluator::get_quadrature_context (
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & integration_point) [nodiscard]

10.254.2.7 invalidate_cache()

void serif::mapping::GridFunctionMappingEvaluator::invalidate_cache () [noexcept]

10.254.2.8 invalidate_for_changed_spaces()

bool serif::mapping::GridFunctionMappingEvaluator::invalidate_for_changed_spaces () [nodiscard], [private]

10.254.2.9 load_element()

void serif::mapping::GridFunctionMappingEvaluator::load_element (
int element_id) [private]

10.254.2.10 refresh()

void serif::mapping::GridFunctionMappingEvaluator::refresh ()

10.254.2.11 validate_field_bindings()

void serif::mapping::GridFunctionMappingEvaluator::validate_field_bindings () const [private]

10.254.3 Member Data Documentation

10.254.3.1 m_cached_element_id

int serif::mapping::GridFunctionMappingEvaluator::m_cached_element_id {-1} [private]

10.254.3.2 m_compactification_coordinate

const mfem::GridFunction& serif::mapping::GridFunctionMappingEvaluator::m_compactification_coordinate [private]

10.254.3.3 m_compactification_data

std::unique_ptr<ElementCompactificationData> serif::mapping::GridFunctionMappingEvaluator::m_compactification_data [private]

10.254.3.4 m_compactification_dofs

mfem::Array<int> serif::mapping::GridFunctionMappingEvaluator::m_compactification_dofs [private]

10.254.3.5 m_compactification_space

const mfem::FiniteElementSpace* serif::mapping::GridFunctionMappingEvaluator::m_compactification_space [private]

10.254.3.6 m_compactification_space_sequence

long serif::mapping::GridFunctionMappingEvaluator::m_compactification_space_sequence [private]

10.254.3.7 m_displacement

const mfem::GridFunction& serif::mapping::GridFunctionMappingEvaluator::m_displacement [private]

10.254.3.8 m_displacement_data

std::unique_ptr<ElementDisplacementData> serif::mapping::GridFunctionMappingEvaluator::m_displacement_data [private]

10.254.3.9 m_displacement_dofs

mfem::Array<int> serif::mapping::GridFunctionMappingEvaluator::m_displacement_dofs [private]

10.254.3.10 m_displacement_space

const mfem::FiniteElementSpace* serif::mapping::GridFunctionMappingEvaluator::m_displacement_space [private]

10.254.3.11 m_displacement_space_sequence

long serif::mapping::GridFunctionMappingEvaluator::m_displacement_space_sequence [private]

10.254.3.12 m_element_compactification

mfem::Vector serif::mapping::GridFunctionMappingEvaluator::m_element_compactification [private]

10.254.3.13 m_element_displacement

mfem::Vector serif::mapping::GridFunctionMappingEvaluator::m_element_displacement [private]

10.254.3.14 m_mapper

const DomainMapper& serif::mapping::GridFunctionMappingEvaluator::m_mapper [private]

10.254.3.15 m_workspace

DMWorkspace serif::mapping::GridFunctionMappingEvaluator::m_workspace [private]

The documentation for this class was generated from the following files:

- src/include/serif/mapping/evaluator.hpp
- src/lib/serif/mapping/evaluator.cpp

10.255 serif::utils::dependencies::detail::HasBinding< BindingsT, KeyT > Struct Template Reference

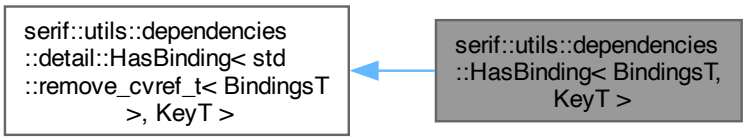
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/binding.hpp

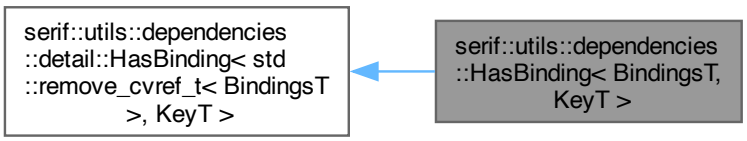
10.256 serif::utils::dependencies::HasBinding< BindingsT, KeyT > Struct Template Reference

#include <binding.hpp>

Inheritance diagram for serif::utils::dependencies::HasBinding< BindingsT, KeyT >:



Collaboration diagram for serif::utils::dependencies::HasBinding< BindingsT, KeyT >:



The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/binding.hpp

10.257 serif::utils::dependencies::detail::HasBinding< Bindings< BindingTs... >, KeyT > Struct Template Reference

#include <binding.hpp>

Static Public Attributes

- static constexpr bool value = (std::is_same_v<KeyT, typename BindingTs::Key> || ...)

10.257.1 Member Data Documentation

10.257.1.1 value

template<typename KeyT, IsBinding... BindingTs>
bool serif::utils::dependencies::detail::HasBinding< Bindings< BindingTs... >, KeyT >::value = (std::is_same_v<KeyT, typename BindingTs::Key> || ...) [static], [constexpr]

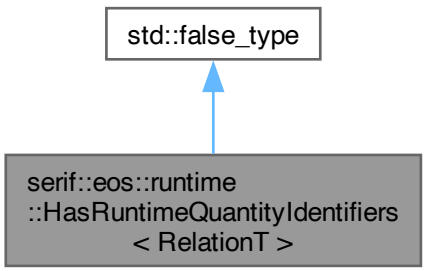
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/binding.hpp

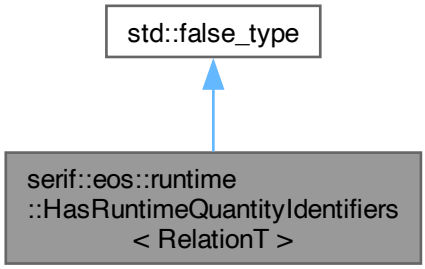
10.258 serif::eos::runtime::HasRuntimeQuantityIdentifiers< RelationT > Struct Template Reference

#include <details.hpp>

Inheritance diagram for serif::eos::runtime::HasRuntimeQuantityIdentifiers< RelationT >:



Collaboration diagram for serif::eos::runtime::HasRuntimeQuantityIdentifiers< RelationT >:



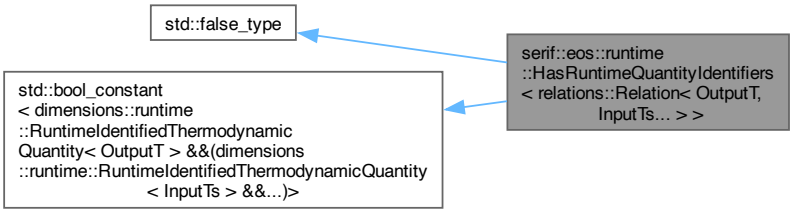
The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

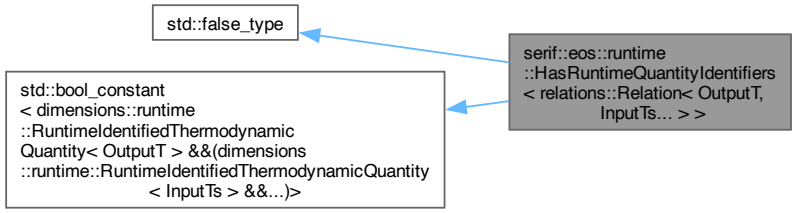
10.259 serif::eos::runtime::HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > > Struct Template Reference

#include <details.hpp>

Inheritance diagram for serif::eos::runtime::HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > >:



Collaboration diagram for serif::eos::runtime::HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > >:



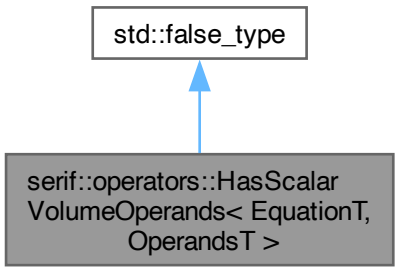
The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

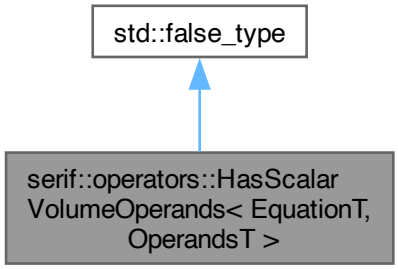
10.260 serif::operators::HasScalarVolumeOperands< EquationT, OperandsT > Struct Template Reference

#include <concepts.hpp>

Inheritance diagram for serif::operators::HasScalarVolumeOperands< EquationT, OperandsT >:



Collaboration diagram for serif::operators::HasScalarVolumeOperands< EquationT, OperandsT >:



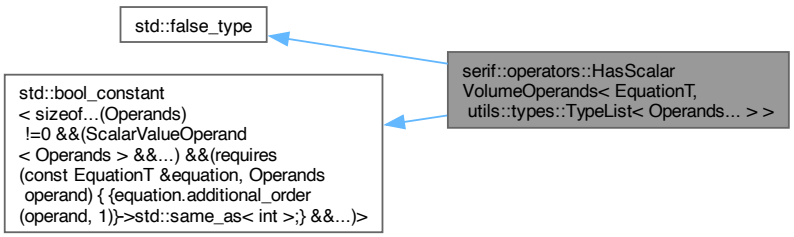
The documentation for this struct was generated from the following file:

- src/include/serif/operators/concepts.hpp

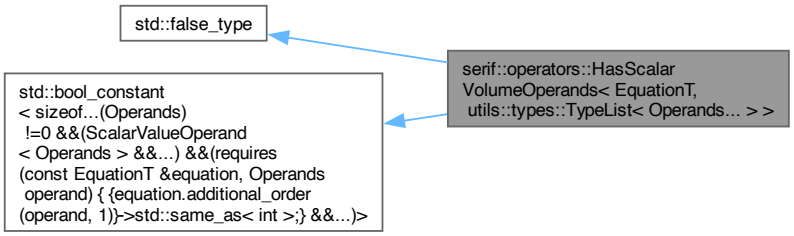
10.261 serif::operators::HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > > Struct Template Reference

#include <concepts.hpp>

Inheritance diagram for serif::operators::HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > >:



Collaboration diagram for serif::operators::HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > >:



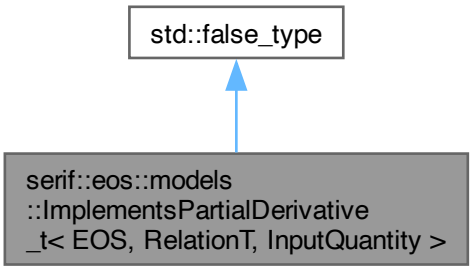
The documentation for this struct was generated from the following file:

- src/include/serif/operators/concepts.hpp

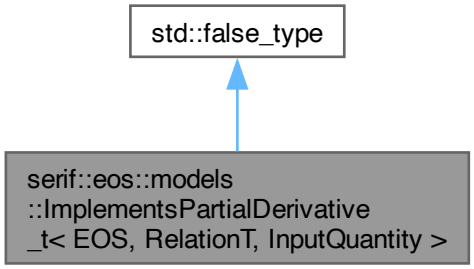
10.262 serif::eos::models::ImplementsPartialDerivative_t< EOS, RelationT, InputQuantity > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::models::ImplementsPartialDerivative_t< EOS, RelationT, InputQuantity >:



Collaboration diagram for serif::eos::models::ImplementsPartialDerivative_t< EOS, RelationT, Input↩ Quantity >:



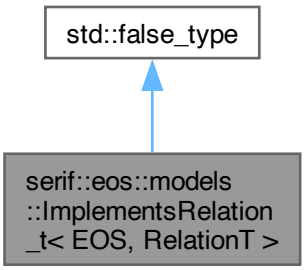
The documentation for this struct was generated from the following file:

- src/include/serif/eos/models/traits.hpp

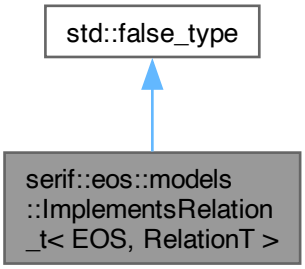
10.263 serif::eos::models::ImplementsRelation_t< EOS, RelationT > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::models::ImplementsRelation_t< EOS, RelationT >:



Collaboration diagram for serif::eos::models::ImplementsRelation_t< EOS, RelationT >:



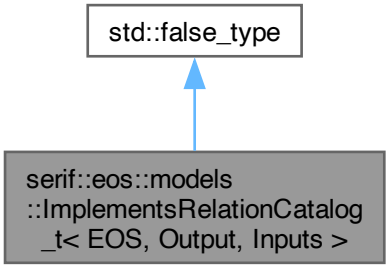
The documentation for this struct was generated from the following file:

- src/include/serif/eos/models/traits.hpp

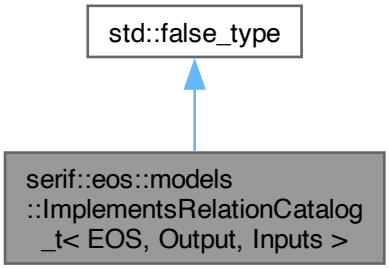
10.264 serif::eos::models::ImplementsRelationCatalog_t< EOS, Output, Inputs > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::models::ImplementsRelationCatalog_t< EOS, Output, Inputs >:



Collaboration diagram for serif::eos::models::ImplementsRelationCatalog_t< EOS, Output, Inputs >:



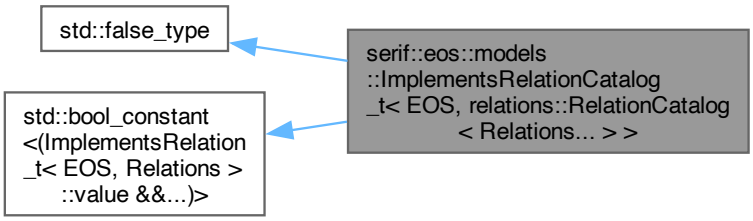
The documentation for this struct was generated from the following file:

- src/include/serif/eos/models/traits.hpp

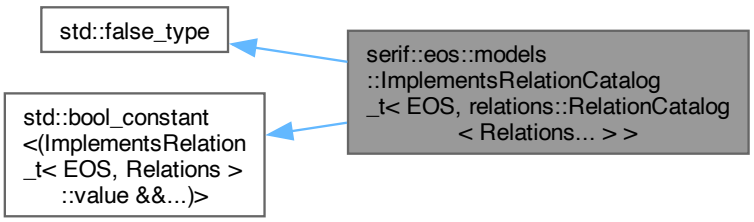
10.265 serif::eos::models::ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::models::ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > >:



Collaboration diagram for serif::eos::models::ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > >:



The documentation for this struct was generated from the following file:

- src/include/serif/eos/models/traits.hpp

10.266 serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT > Class Template Reference

A calculation may reuse unchanged pieces of its previous immutable result.

#include <incremental_computation.hpp>

Public Types

- using Key = KeyT
- using Inputs = std::remove_cvref_t<InputsT>
- using Function = FunctionT
- using Update = ComputationUpdate<Inputs, ValueT>

Public Member Functions

- IncrementalComputation (FunctionT function)

  Own a callable taking (update, declared inputs...) and returning Value or expected<Value, Error>.
- template<typename... Arguments>
  requires std::invocable<const FunctionT&, const Update&, const Arguments&...> && std::same_as<typename detail::Result<std::remove_cv_t<std::invoke_result_t<const FunctionT&, const Update&, const Arguments&...>>>::Value, ValueT>
  decltype(auto) evaluate (const Arguments &... arguments) const

  Without a cache, evaluation is a complete calculation with no previous result.
- template<typename... Arguments>
  decltype(auto) evaluate_update (const Update &update, const Arguments &... arguments) const

  Previous state is read-only; only the returned candidate can become the next accepted result.

Private Attributes

- FunctionT m_function

10.266.1 Detailed Description

template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
requires std::is_object_v<ValueT> && std::move_constructible<ValueT>
class serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >

A calculation may reuse unchanged pieces of its previous immutable result.

10.266.2 Member Typedef Documentation

10.266.2.1 Function

template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
using serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >::Function = FunctionT

10.266.2.2 Inputs

template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
using serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >::Inputs = std::remove_cvref↩
_t<InputsT>

10.266.2.3 Key

template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
using serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >::Key = KeyT

10.266.2.4 Update

template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
using serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >::Update = ComputationUpdate<Inputs,
ValueT>

10.266.3 Constructor & Destructor Documentation

10.266.3.1 IncrementalComputation()

template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >::IncrementalComputation (
FunctionT function) [inline], [explicit]

Own a callable taking (update, declared inputs...) and returning Value or expected<Value, Error>.

10.266.4 Member Function Documentation

10.266.4.1 evaluate()

template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
template<typename... Arguments>
requires std::invocable<const FunctionT&, const Update&, const Arguments&...> && std::same_as<typename detail::Result<std::remove_cv_t<std::invoke_result_t<const FunctionT&, const Update&, const Arguments&...>>>::Value, ValueT>
decltype(auto) serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >::evaluate (
const Arguments &... arguments) const [inline], [nodiscard]

Without a cache, evaluation is a complete calculation with no previous result.

10.266.4.2 evaluate_update()

template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
template<typename... Arguments>
decltype(auto) serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >::evaluate_update (
const Update & update,
const Arguments &... arguments) const [inline], [nodiscard]

Previous state is read-only; only the returned candidate can become the next accepted result.

10.266.5 Member Data Documentation

10.266.5.1 m_function

template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
FunctionT serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >::m_function [private]

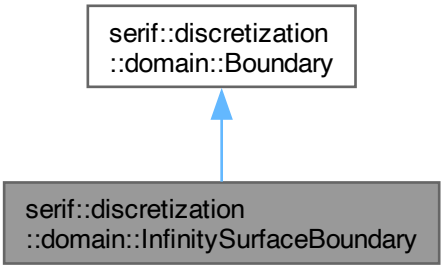
The documentation for this class was generated from the following file:

- src/include/serif/utils/dependencies/incremental_computation.hpp

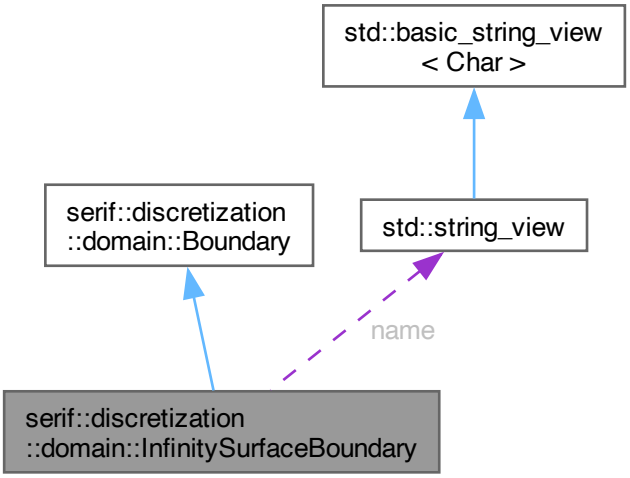
10.267 serif::discretization::domain::InfinitySurfaceBoundary Struct Reference

#include <types.hpp>

Inheritance diagram for serif::discretization::domain::InfinitySurfaceBoundary:



Collaboration diagram for serif::discretization::domain::InfinitySurfaceBoundary:



Static Public Attributes

- static constexpr std::string_view name = "infinity_surface"

10.267.1 Member Data Documentation

10.267.1.1 name

std::string_view serif::discretization::domain::InfinitySurfaceBoundary::name = "infinity_surface" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/types.hpp

10.268 serif::utils::dependencies::keys::Input Struct Reference

The main input to a computation.

#include <keys.hpp>

10.268.1 Detailed Description

The main input to a computation.

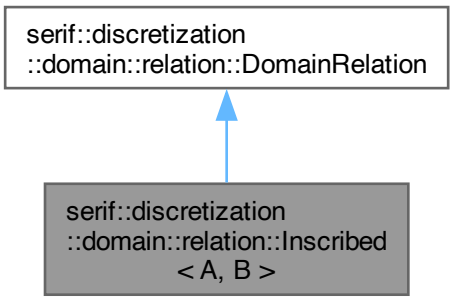
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/keys.hpp

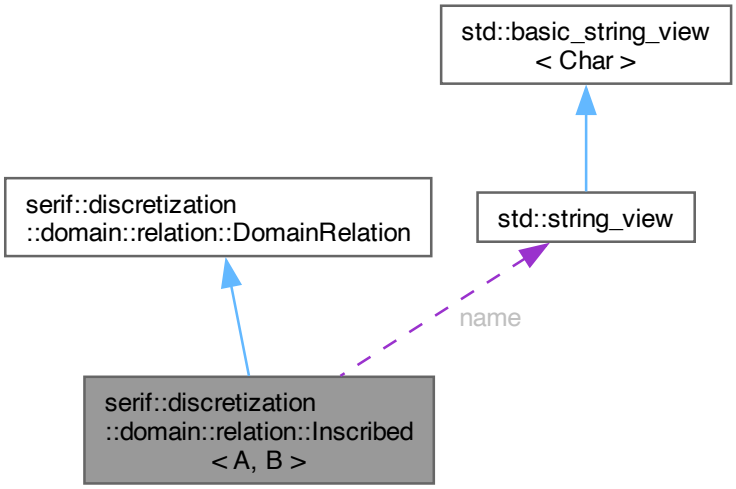
10.269 serif::discretization::domain::relation::Inscribed< A, B > Struct Template Reference

#include <relations.hpp>

Inheritance diagram for serif::discretization::domain::relation::Inscribed< A, B >:



Collaboration diagram for serif::discretization::domain::relation::Inscribed< A, B >:



Public Types

- using inner_type = A
- using outer_type = B

Static Public Attributes

- static constexpr std::string_view name = "inscribed"

10.269.1 Member Typedef Documentation

10.269.1.1 inner_type

template<IsDomainOrSet A, IsDomainOrSet B>
using serif::discretization::domain::relation::Inscribed< A, B >::inner_type = A

10.269.1.2 outer_type

template<IsDomainOrSet A, IsDomainOrSet B>
using serif::discretization::domain::relation::Inscribed< A, B >::outer_type = B

10.269.2 Member Data Documentation

10.269.2.1 name

template<IsDomainOrSet A, IsDomainOrSet B>
std::string_view serif::discretization::domain::relation::Inscribed< A, B >::name = "inscribed" [static], [constexpr]

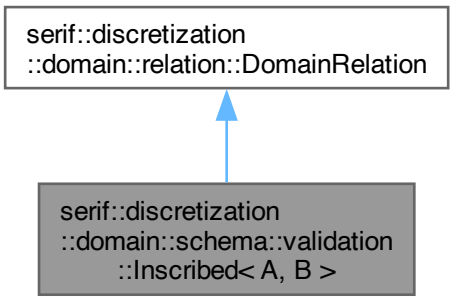
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/relations.hpp

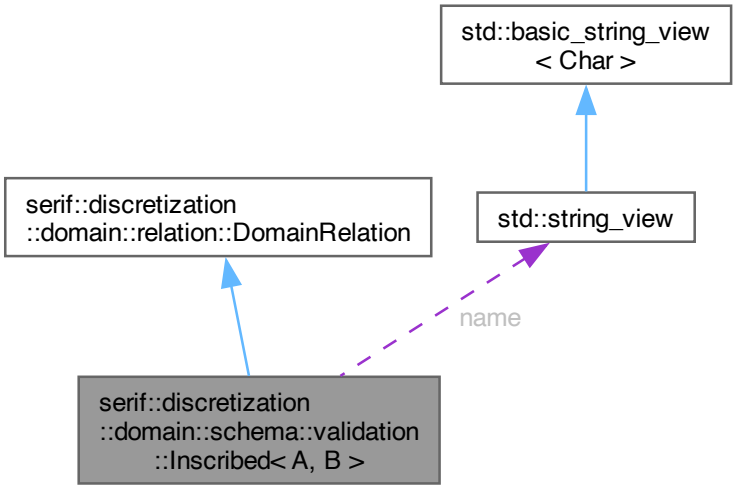
10.270 serif::discretization::domain::schema::validation::Inscribed< A, B > Struct Template Reference

#include <relations.hpp>

Inheritance diagram for serif::discretization::domain::schema::validation::Inscribed< A, B >:



Collaboration diagram for serif::discretization::domain::schema::validation::Inscribed< A, B >:



Public Types

- using inner_type = A
- using outer_type = B

Static Public Attributes

- static constexpr std::string_view name = "inscribed"

10.270.1 Member Typedef Documentation

10.270.1.1 inner_type

template<IsDomainOrSet A, IsDomainOrSet B>
using serif::discretization::domain::relation::Inscribed< A, B >::inner_type = A

10.270.1.2 outer_type

template<IsDomainOrSet A, IsDomainOrSet B>
using serif::discretization::domain::relation::Inscribed< A, B >::outer_type = B

10.270.2 Member Data Documentation

10.270.2.1 name

template<IsDomainOrSet A, IsDomainOrSet B>
std::string_view serif::discretization::domain::relation::Inscribed< A, B >::name = "inscribed" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/relations.hpp

10.271 serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics Struct Reference

#include <runtime.hpp>

Public Attributes

- int faceID {-1}
- int innerElementID {-1}
- int adjacentElementID {-1}
- int adjacentDomainID {-1}

10.271.1 Member Data Documentation

10.271.1.1 adjacentDomainID

int serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics::adjacentDomainID {-1}

10.271.1.2 adjacentElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics::adjacentElementID {-1}

10.271.1.3 faceID

int serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics::faceID {-1}

10.271.1.4 innerElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics::innerElementID {-1}

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/runtime.hpp

10.272 serif::discretization::domain::schema::validation::Relation↩ ValidationResult::InscribedDiagnostics Struct Reference

#include <runtime.hpp>

Public Attributes

- int faceID {-1}
- int innerElementID {-1}
- int adjacentElementID {-1}
- int adjacentDomainID {-1}

10.272.1 Member Data Documentation

10.272.1.1 adjacentDomainID

int serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics::adjacentDomainID {-1}

10.272.1.2 adjacentElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics::adjacentElementID {-1}

10.272.1.3 faceID

int serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics::faceID {-1}

10.272.1.4 innerElementID

int serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics::innerElementID {-1}

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/runtime.hpp

10.273 serif::geometry::deformation::keys::InteriorModel Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

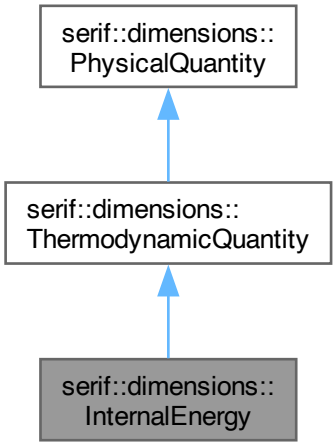
- src/include/serif/geometry/deformation/computations.hpp

10.274 serif::dimensions::InternalEnergy Struct Reference

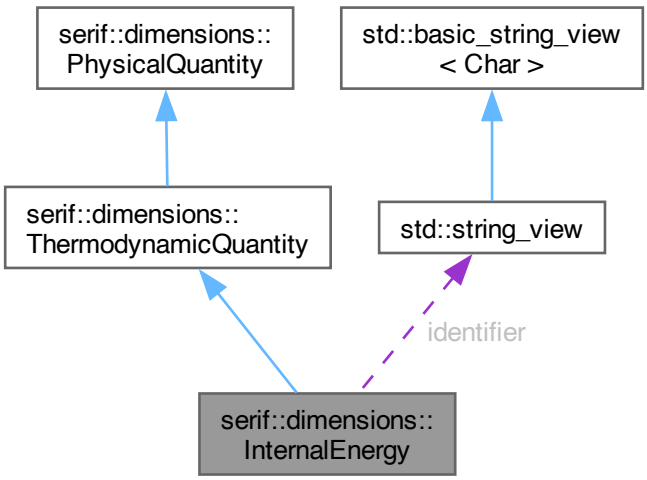
Dimension of Internal Energy.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::InternalEnergy:



Collaboration diagram for serif::dimensions::InternalEnergy:



Static Public Attributes

- static constexpr std::string_view identifier = ”internal_energy”

10.274.1 Detailed Description

Dimension of Internal Energy.

10.274.2 Member Data Documentation

10.274.2.1 identifier

std::string_view serif::dimensions::InternalEnergy::identifier = ”internal_energy” [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.275 serif::utils::dependencies::detail::Invocation< ComputationT, ArgumentsT, typename > Struct Template Reference

#include <result.hpp>

Static Public Attributes

- static constexpr bool valid = false

10.275.1 Member Data Documentation

10.275.1.1 valid

template<typename ComputationT, typename ArgumentsT, typename = void>
bool serif::utils::dependencies::detail::Invocation< ComputationT, ArgumentsT, typename >::valid = false [static], [constexpr]

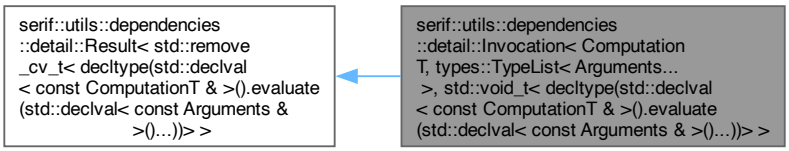
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/result.hpp

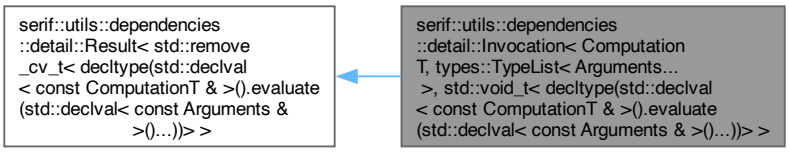
10.276 serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< decltype(std::declval< const ComputationT & >().evaluate(std::declval< const Arguments & >()...))> > Struct Template Reference

#include <result.hpp>

Inheritance diagram for serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< decltype(std::declval< const ComputationT & >().evaluate(std::declval< const Arguments & >()...))> >:



Collaboration diagram for serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< decltype(std::declval< const ComputationT & >().evaluate(std::declval< const Arguments & >()...))> >:



Static Public Attributes

- static constexpr bool valid

Static Public Attributes inherited from serif::utils::dependencies::detail::Result< std::remove_cv_t< decltype(std::declval< const ComputationT

- static constexpr bool fallible
- static constexpr bool valid

Additional Inherited Members

Public Types inherited from serif::utils::dependencies::detail::Result< std::remove_cv_t< decltype(std::declval< const ComputationT

- using Value
- using Error

10.276.1 Member Data Documentation

10.276.1.1 valid

bool serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, typename >::valid [static], [constexpr]

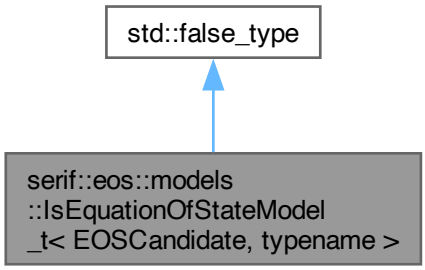
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/result.hpp

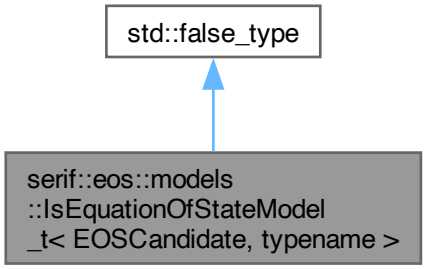
10.277 serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, typename > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, typename >:



Collaboration diagram for serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, typename >:



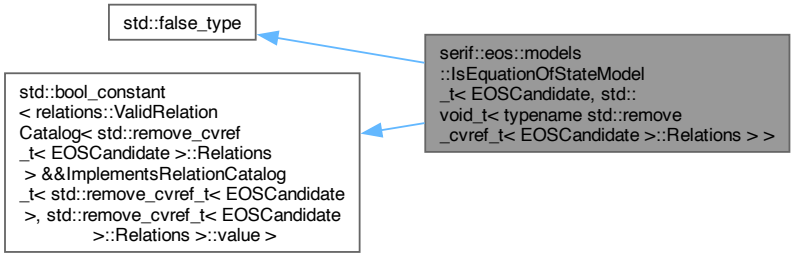
The documentation for this struct was generated from the following file:

- src/include/serif/eos/models/traits.hpp

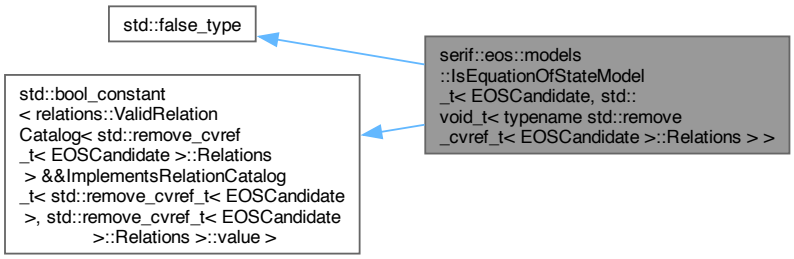
10.278 serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std::remove_cvref_t< EOSCandidate >::Relations > > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std::remove_cvref_t< EOSCandidate >::Relations > >:



Collaboration diagram for serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std::remove_cvref_t< EOSCandidate >::Relations > >:



The documentation for this struct was generated from the following file:

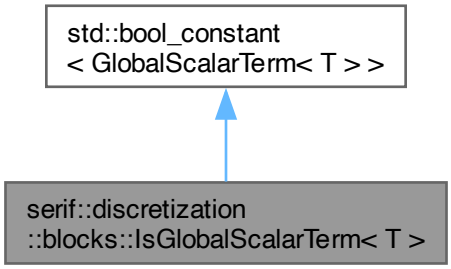
- src/include/serif/eos/models/traits.hpp

10.279 serif::discretization::blocks::IsGlobalScalarTerm< T > Struct Template Reference

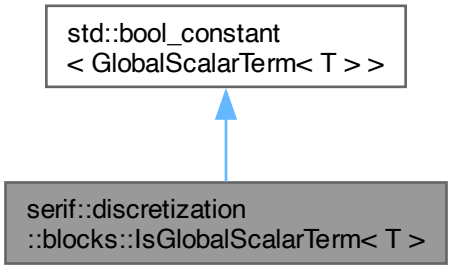
Helper struct to determine if a type is a global scalar term (GlobalScalarTerm).

#include <concepts.hpp>

Inheritance diagram for serif::discretization::blocks::IsGlobalScalarTerm< T >:



Collaboration diagram for serif::discretization::blocks::IsGlobalScalarTerm< T >:



10.279.1 Detailed Description

template<typename T>
struct serif::discretization::blocks::IsGlobalScalarTerm< T >

Helper struct to determine if a type is a global scalar term (GlobalScalarTerm).

Template Parameters

T	The type to check.

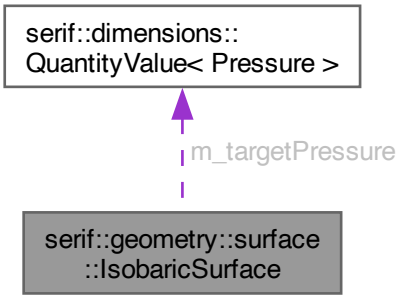
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/concepts.hpp

10.280 serif::geometry::surface::IsobaricSurface Class Reference

#include <isobaric.hpp>

Collaboration diagram for serif::geometry::surface::IsobaricSurface:



Classes

- struct Parameters

Public Types

- using PhysicalQuantity = dimensions::Pressure
- using TargetValue = dimensions::PressureValue

Public Member Functions

- IsobaricSurface (const Parameters parameters)
- IsobaricSurface (const TargetValue targetPressure)
- TargetValue target_pressure () const noexcept
- std::array< double, 1 > continuation_identity () const noexcept
- PressureSurfaceDescriptor descriptor () const noexcept

Private Attributes

- TargetValue m_targetPressure

10.280.1 Member Typedef Documentation

10.280.1.1 PhysicalQuantity

using serif::geometry::surface::IsobaricSurface::PhysicalQuantity = dimensions::Pressure

10.280.1.2 TargetValue

using serif::geometry::surface::IsobaricSurface::TargetValue = dimensions::PressureValue

10.280.2 Constructor & Destructor Documentation

10.280.2.1 IsobaricSurface() [1/2]

serif::geometry::surface::IsobaricSurface::IsobaricSurface (
const Parameters parameters) [explicit]

10.280.2.2 IsobaricSurface() [2/2]

serif::geometry::surface::IsobaricSurface::IsobaricSurface (
const TargetValue targetPressure) [explicit]

10.280.3 Member Function Documentation

10.280.3.1 continuation_identity()

std::array< double, 1 > serif::geometry::surface::IsobaricSurface::continuation_identity () const [nodiscard], [noexcept]

10.280.3.2 descriptor()

PressureSurfaceDescriptor serif::geometry::surface::IsobaricSurface::descriptor () const [nodiscard], [noexcept]

10.280.3.3 target_pressure()

IsobaricSurface::TargetValue serif::geometry::surface::IsobaricSurface::target_pressure () const [nodiscard], [noexcept]

10.280.4 Member Data Documentation

10.280.4.1 m_targetPressure

TargetValue serif::geometry::surface::IsobaricSurface::m_targetPressure [private]

The documentation for this class was generated from the following files:

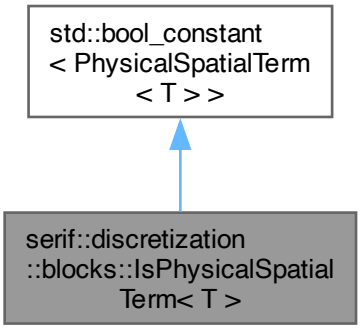
- src/include/serif/geometry/surface/isobaric.hpp
- src/lib/serif/geometry/surface/isobaric.cpp

10.281 serif::discretization::blocks::IsPhysicalSpatialTerm< T > Struct Template Reference

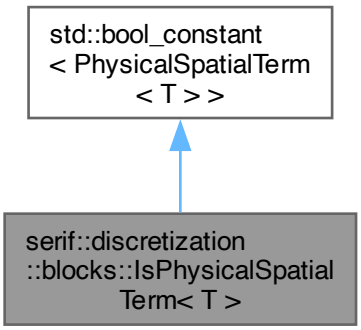
Helper struct to determine if a type is a physical spatial term (PhysicalSpatialTerm).

#include <concepts.hpp>

Inheritance diagram for serif::discretization::blocks::IsPhysicalSpatialTerm< T >:



Collaboration diagram for serif::discretization::blocks::IsPhysicalSpatialTerm< T >:



10.281.1 Detailed Description

template<typename T>
struct serif::discretization::blocks::IsPhysicalSpatialTerm< T >

Helper struct to determine if a type is a physical spatial term (PhysicalSpatialTerm).

Template Parameters

T	The type to check.

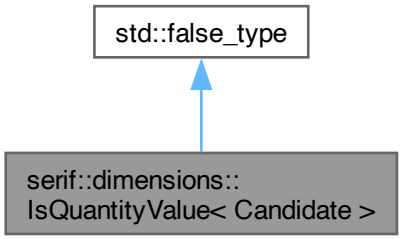
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/concepts.hpp

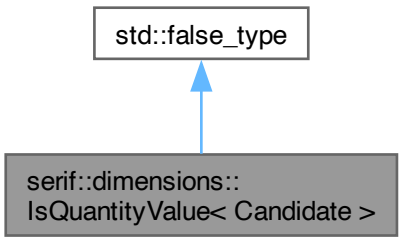
10.282 serif::dimensions::IsQuantityValue< Candidate > Struct Template Reference

#include <quantities.hpp>

Inheritance diagram for serif::dimensions::IsQuantityValue< Candidate >:



Collaboration diagram for serif::dimensions::IsQuantityValue< Candidate >:



10.282.1 Detailed Description

template<typename Candidate>
struct serif::dimensions::IsQuantityValue< Candidate >

Template Parameters

Candidate	some type that may be a QuantityValue

@breif Most general specialization, defaults to compile time false.

Template Parameters

Quantity	Some quantity that satisfies the PhysicalQuantityType concept

@breif More specific so if a type satisfies this the compiler will select it first. This evaluates to compile time true if Quantity satisfies the PhysicalQuantityType concept which then creats a specialization of IsQuantityValue that is compile time true for any Quantity of that set.

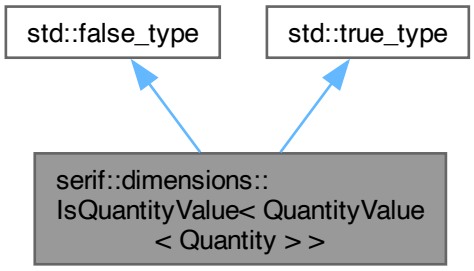
The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/quantities.hpp

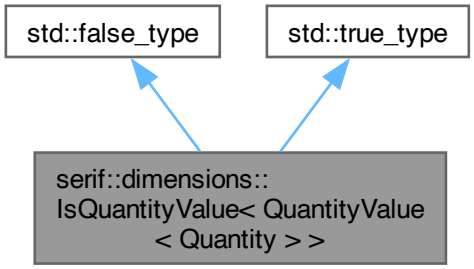
10.283 serif::dimensions::IsQuantityValue< QuantityValue< Quantity > > Struct Template Reference

#include <quantities.hpp>

Inheritance diagram for serif::dimensions::IsQuantityValue< QuantityValue< Quantity > >:



Collaboration diagram for serif::dimensions::IsQuantityValue< QuantityValue< Quantity > >:



The documentation for this struct was generated from the following file:

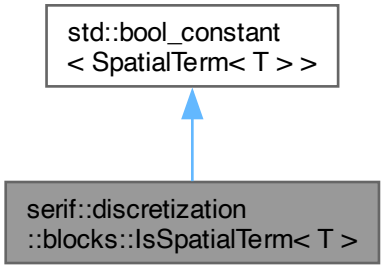
- src/include/serif/dimensions/quantities.hpp

10.284 serif::discretization::blocks::IsSpatialTerm< T > Struct Template Reference

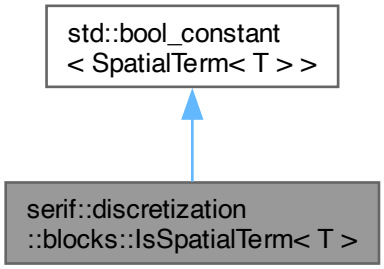
Helper struct to determine if a type is a spatial term (SpatialTerm).

#include <concepts.hpp>

Inheritance diagram for serif::discretization::blocks::IsSpatialTerm< T >:



Collaboration diagram for serif::discretization::blocks::IsSpatialTerm< T >:



10.284.1 Detailed Description

template<typename T>
struct serif::discretization::blocks::IsSpatialTerm< T >

Helper struct to determine if a type is a spatial term (SpatialTerm).

Template Parameters

T	The type to check.

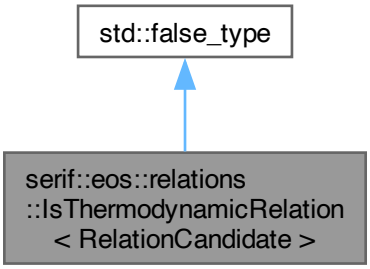
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/concepts.hpp

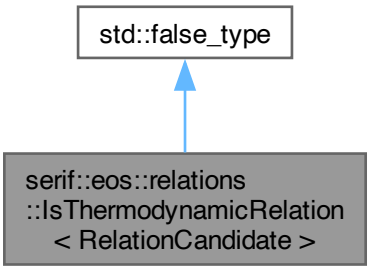
10.285 serif::eos::relations::IsThermodynamicRelation< RelationCandidate > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::relations::IsThermodynamicRelation< RelationCandidate >:



Collaboration diagram for serif::eos::relations::IsThermodynamicRelation< RelationCandidate >:



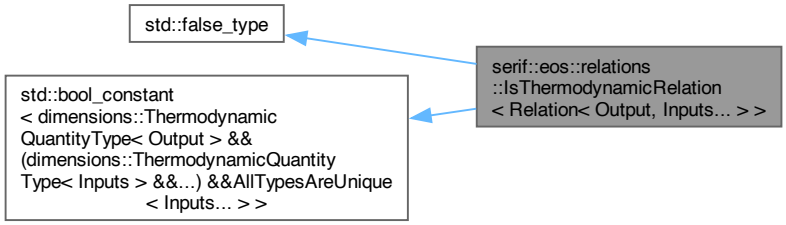
The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/traits.hpp

10.286 serif::eos::relations::IsThermodynamicRelation< Relation< Output, Inputs... > > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::relations::IsThermodynamicRelation< Relation< Output, Inputs... > >:



Collaboration diagram for serif::eos::relations::IsThermodynamicRelation< Relation< Output, Inputs... > >:

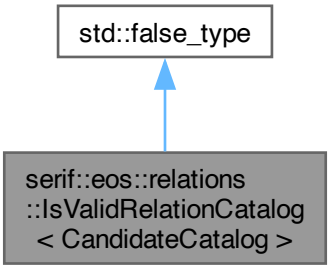
The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/traits.hpp

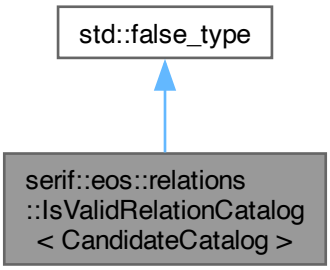
10.287 serif::eos::relations::IsValidRelationCatalog< CandidateCatalog > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::relations::IsValidRelationCatalog< CandidateCatalog >:



Collaboration diagram for serif::eos::relations::IsValidRelationCatalog< CandidateCatalog >:



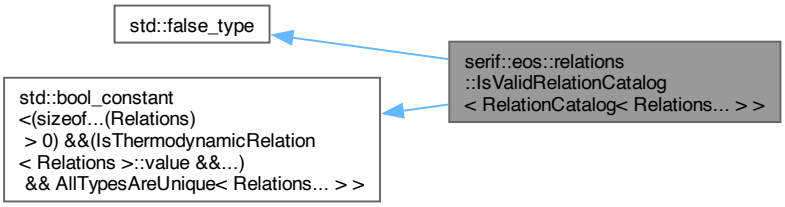
The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/traits.hpp

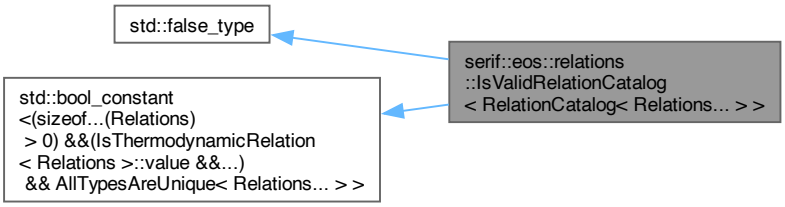
10.288 serif::eos::relations::IsValidRelationCatalog< RelationCatalog< Relations... > > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::relations::IsValidRelationCatalog< RelationCatalog< Relations... > >:



Collaboration diagram for serif::eos::relations::IsValidRelationCatalog< RelationCatalog< Relations... > >:



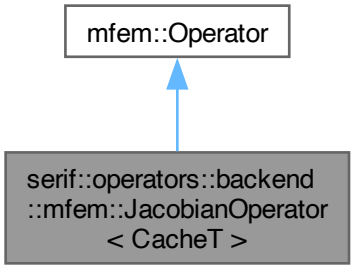
The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/traits.hpp

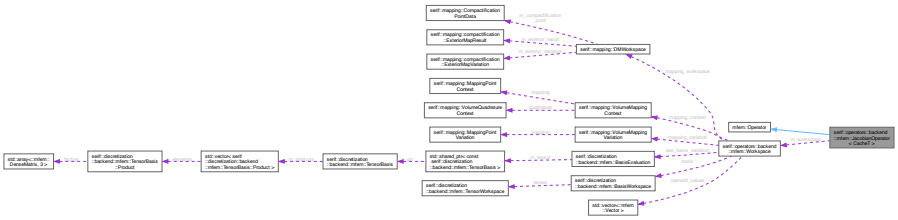
10.289 serif::operators::backend::mfem::JacobianOperator< CacheT > Class Template Reference

#include <jacobian_operator.hpp>

Inheritance diagram for serif::operators::backend::mfem::JacobianOperator< CacheT >:



Collaboration diagram for serif::operators::backend::mfem::JacobianOperator< CacheT >:



Public Member Functions

- JacobianOperator (const CacheT &cache)

  Fix dimensions from an already accepted volume linearization.
- JacobianOperator (CacheT &&)=delete
- JacobianOperator (const CacheT &&)=delete
- void Mult (const ::mfem::Vector &direction, ::mfem::Vector &result) const override

  Apply the latest accepted Jacobian; a failed preparation makes this unavailable.
- void MultTranspose (const ::mfem::Vector &dual, ::mfem::Vector &result) const override

  Apply the coordinate transpose, including operand, test and geometry variations.

Private Member Functions

- const auto & checked_data () const

  MFEM's fixed dimensions must still match the cache's current accepted result.

Private Attributes

- const CacheT & m_cache
- Workspace m_workspace

10.289.1 Detailed Description

template<typename CacheT>
class serif::operators::backend::mfem::JacobianOperator< CacheT >

Adapt an accepted dependency cache directly to MFEM's linear operator interface. The cache must outlive this adapter. Calls sharing spaces or a mapper must be serialized.

10.289.2 Constructor & Destructor Documentation

10.289.2.1 JacobianOperator() [1/3]

template<typename CacheT>
serif::operators::backend::mfem::JacobianOperator< CacheT >::JacobianOperator (
const CacheT & cache) [inline], [explicit]

Fix dimensions from an already accepted volume linearization.

10.289.2.2 JacobianOperator() [2/3]

template<typename CacheT>
serif::operators::backend::mfem::JacobianOperator< CacheT >::JacobianOperator (
CacheT &&) [delete]

10.289.2.3 JacobianOperator() [3/3]

template<typename CacheT>
serif::operators::backend::mfem::JacobianOperator< CacheT >::JacobianOperator (
const CacheT &&) [delete]

10.289.3 Member Function Documentation

10.289.3.1 checked_data()

template<typename CacheT>
const auto & serif::operators::backend::mfem::JacobianOperator< CacheT >::checked_data () const [inline], [nodiscard], [private]

MFEM's fixed dimensions must still match the cache's current accepted result.

10.289.3.2 Mult()

template<typename CacheT>
void serif::operators::backend::mfem::JacobianOperator< CacheT >::Mult (
const ::mfem::Vector & direction,
::mfem::Vector & result) const [inline], [override]

Apply the latest accepted Jacobian; a failed preparation makes this unavailable.

10.289.3.3 MultTranspose()

template<typename CacheT>
void serif::operators::backend::mfem::JacobianOperator< CacheT >::MultTranspose (
const ::mfem::Vector & dual,
::mfem::Vector & result) const [inline], [override]

Apply the coordinate transpose, including operand, test and geometry variations.

10.289.4 Member Data Documentation

10.289.4.1 m_cache

template<typename CacheT>
const CacheT& serif::operators::backend::mfem::JacobianOperator< CacheT >::m_cache [private]

10.289.4.2 m_workspace

template<typename CacheT>
Workspace serif::operators::backend::mfem::JacobianOperator< CacheT >::m_workspace [mutable], [private]

The documentation for this class was generated from the following file:

- src/include/serif/operators/backend/mfem/jacobian_operator.hpp

10.290 serif::operators::detail::Join< Lists > Struct Template Reference

#include <plan.hpp>

Public Types

- using type = typename utils::dependencies::CombinedKeyList<typename AsKeyList<Lists>::type...>::Keys

10.290.1 Member Typedef Documentation

10.290.1.1 type

template<typename... Lists>
using serif::operators::detail::Join< Lists >::type = typename utils::dependencies::CombinedKeyList<typename AsKeyList<Lists>::type...>::Keys

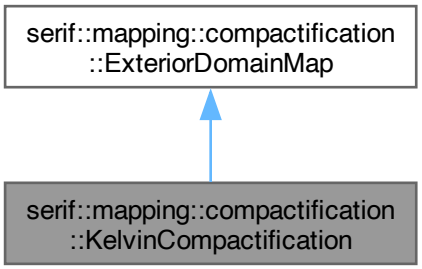
The documentation for this struct was generated from the following file:

- src/include/serif/operators/plan.hpp

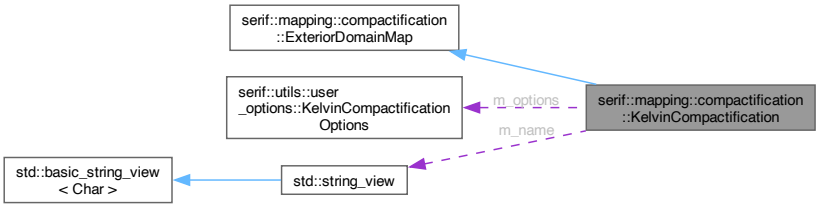
10.291 serif::mapping::compactification::KelvinCompactification Class Reference

#include <kelvin.hpp>

Inheritance diagram for serif::mapping::compactification::KelvinCompactification:



Collaboration diagram for serif::mapping::compactification::KelvinCompactification:



Classes

- struct RadialFactors

Public Member Functions

- KelvinCompactification (const utils::user_options::KelvinCompactificationOptions &options)
- bool is_affine_in_displacement () const noexcept override
- MappingStatus evaluate (const ExteriorMapInput &input, ExteriorMapResult &result) const override
- MappingStatus evaluate_variation (const ExteriorMapInput &input, const ExteriorMapResult &result, const ExteriorMapDirection &direction, ExteriorMapVariation &variation) const override
- std::string_view get_name () const noexcept override
- std::unique_ptr< const ExteriorDomainMap > clone () const override
- double get_reference_stellar_radius () const noexcept
- double get_reference_infinity_radius () const noexcept
- double get_coordinate_tolerance () const noexcept
- MappingStatus evaluate_jacobian_variation_factors (double compactification_coordinate, const mfem::Vector &coordinateGradient, double &scale, mfem::Vector &scaleGradient) const

Public Member Functions inherited from serif::mapping::compactification::ExteriorDomainMap

- virtual ~ExteriorDomainMap ()=default

Private Member Functions

- MappingStatus compute_radial_factors (double compactification_coordinate, RadialFactors &radial_factors) const

Private Attributes

- utils::user_options::KelvinCompactificationOptions m_options
- const std::string_view m_name = "KelvinCompactification"

10.291.1 Constructor & Destructor Documentation

10.291.1.1 KelvinCompactification()

serif::mapping::compactification::KelvinCompactification::KelvinCompactification (
const utils::user_options::KelvinCompactificationOptions & options) [explicit]

10.291.2 Member Function Documentation

10.291.2.1 clone()

std::unique_ptr< const ExteriorDomainMap > serif::mapping::compactification::KelvinCompactification::clone () const [nodiscard], [override], [virtual]

Implements serif::mapping::compactification::ExteriorDomainMap.

10.291.2.2 compute_radial_factors()

MappingStatus serif::mapping::compactification::KelvinCompactification::compute_radial_factors (
double compactification_coordinate,
RadialFactors & radial_factors) const [nodiscard], [private]

10.291.2.3 evaluate()

MappingStatus serif::mapping::compactification::KelvinCompactification::evaluate (
const ExteriorMapInput & input,
ExteriorMapResult & result) const [nodiscard], [override], [virtual]

Implements serif::mapping::compactification::ExteriorDomainMap.

10.291.2.4 evaluate_jacobian_variation_factors()

MappingStatus serif::mapping::compactification::KelvinCompactification::evaluate_jacobian_variation_factors (
double compactification_coordinate,
const mfem::Vector & coordinateGradient,
double & scale,
mfem::Vector & scaleGradient) const [nodiscard]

10.291.2.5 evaluate_variation()

MappingStatus serif::mapping::compactification::KelvinCompactification::evaluate_variation (
const ExteriorMapInput & input,
const ExteriorMapResult & result,
const ExteriorMapDirection & direction,
ExteriorMapVariation & variation) const [nodiscard], [override], [virtual]

Implements serif::mapping::compactification::ExteriorDomainMap.

10.291.2.6 get_coordinate_tolerance()

double serif::mapping::compactification::KelvinCompactification::get_coordinate_tolerance () const [nodiscard], [noexcept]

10.291.2.7 get_name()

std::string_view serif::mapping::compactification::KelvinCompactification::get_name () const [nodiscard], [override], [virtual], [noexcept]

Implements serif::mapping::compactification::ExteriorDomainMap.

10.291.2.8 get_reference_infinity_radius()

double serif::mapping::compactification::KelvinCompactification::get_reference_infinity_radius () const [nodiscard], [noexcept]

10.291.2.9 get_reference_stellar_radius()

double serif::mapping::compactification::KelvinCompactification::get_reference_stellar_radius () const [nodiscard], [noexcept]

10.291.2.10 is_affine_in_displacement()

bool serif::mapping::compactification::KelvinCompactification::is_affine_in_displacement () const [nodiscard], [override], [virtual], [noexcept]

Implements serif::mapping::compactification::ExteriorDomainMap.

10.291.3 Member Data Documentation

10.291.3.1 m_name

const std::string_view serif::mapping::compactification::KelvinCompactification::m_name = "KelvinCompactification" [private]

10.291.3.2 m_options

utils::user_options::KelvinCompactificationOptions serif::mapping::compactification::KelvinCompactification::m_options [private]

The documentation for this class was generated from the following files:

- src/include/serif/mapping/compactification/kelvin.hpp
- src/lib/serif/mapping/compactification/kelvin.cpp

10.292 serif::utils::user_options::KelvinCompactificationOptions Struct Reference

#include <kelvin.hpp>

Public Attributes

- double r_star_ref {1.0}
- double r_inf_ref {2.0}
- double coordinate_tolerance {1.0e-12}

10.292.1 Member Data Documentation

10.292.1.1 coordinate_tolerance

double serif::utils::user_options::KelvinCompactificationOptions::coordinate_tolerance {1.0e-12}

10.292.1.2 r_inf_ref

double serif::utils::user_options::KelvinCompactificationOptions::r_inf_ref {2.0}

10.292.1.3 r_star_ref

double serif::utils::user_options::KelvinCompactificationOptions::r_star_ref {1.0}

The documentation for this struct was generated from the following file:

- src/include/serif/utils/user_options/mapping/compactification/kelvin.hpp

10.293 serif::utils::dependencies::KeyList< KeyTs > Struct Template Reference

An ordered collection of role keys, independent of their associated value types.

#include <key_list.hpp>

Public Types

- using Keys = types::TypeList<KeyTs...>

10.293.1 Detailed Description

template<typename... KeyTs>
struct serif::utils::dependencies::KeyList< KeyTs >

An ordered collection of role keys, independent of their associated value types.

10.293.2 Member Typedef Documentation

10.293.2.1 Keys

template<typename... KeyTs>
using serif::utils::dependencies::KeyList< KeyTs >::Keys = types::TypeList<KeyTs...>

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/key_list.hpp

10.294 serif::utils::dependencies::KeyListBindings< BindingsT, KeysT > Struct Template Reference

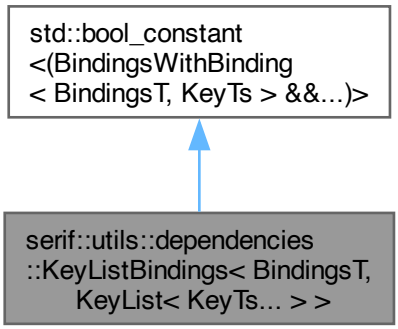
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/key_list.hpp

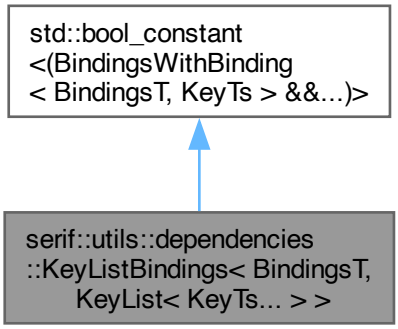
10.295 serif::utils::dependencies::KeyListBindings< BindingsT, KeyList< KeyTs... > > Struct Template Reference

#include <key_list.hpp>

Inheritance diagram for serif::utils::dependencies::KeyListBindings< BindingsT, KeyList< KeyTs... > >:



Collaboration diagram for serif::utils::dependencies::KeyListBindings< BindingsT, KeyList< KeyTs... > >:



The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/key_list.hpp

10.296 serif::utils::dependencies::detail::KeyListSize< KeysT > Struct Template Reference

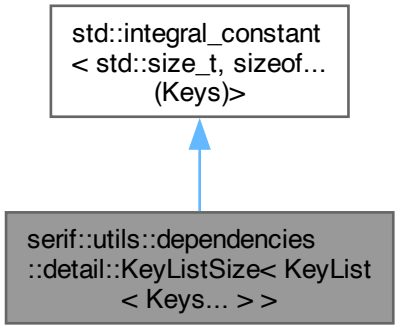
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/type_lists.hpp

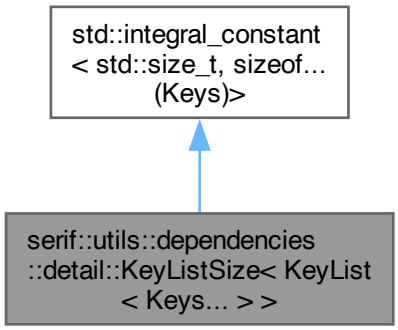
10.297 serif::utils::dependencies::detail::KeyListSize< KeyList< Keys... > > Struct Template Reference

#include <type_lists.hpp>

Inheritance diagram for serif::utils::dependencies::detail::KeyListSize< KeyList< Keys... > >:



Collaboration diagram for serif::utils::dependencies::detail::KeyListSize< KeyList< Keys... > >:



The documentation for this struct was generated from the following file:

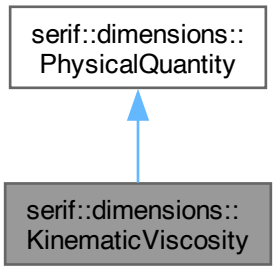
- src/include/serif/utils/dependencies/detail/type_lists.hpp

10.298 serif::dimensions::KinematicViscosity Struct Reference

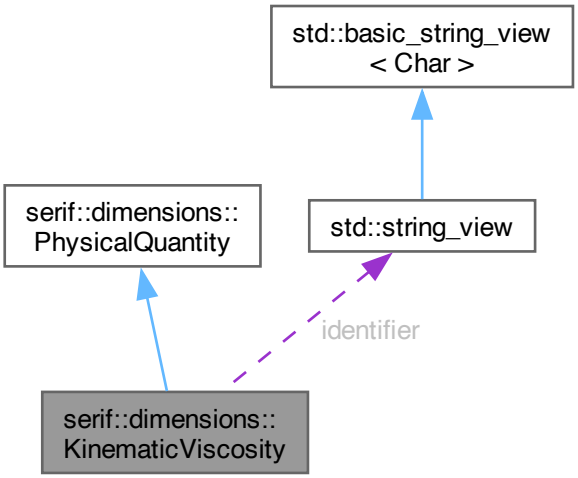
Dimension of Kinematic Viscosity.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::KinematicViscosity:



Collaboration diagram for serif::dimensions::KinematicViscosity:



Static Public Attributes

- static constexpr std::string_view identifier = "kinematic_viscosity"

10.298.1 Detailed Description

Dimension of Kinematic Viscosity.

10.298.2 Member Data Documentation

10.298.2.1 identifier

std::string_view serif::dimensions::KinematicViscosity::identifier = "kinematic_viscosity" [static], [constexpr]

The documentation for this struct was generated from the following file:

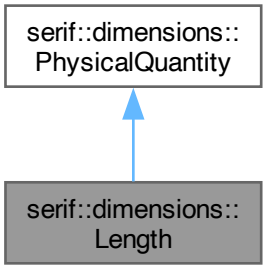
- src/include/serif/dimensions/dimensions.hpp

10.299 serif::dimensions::Length Struct Reference

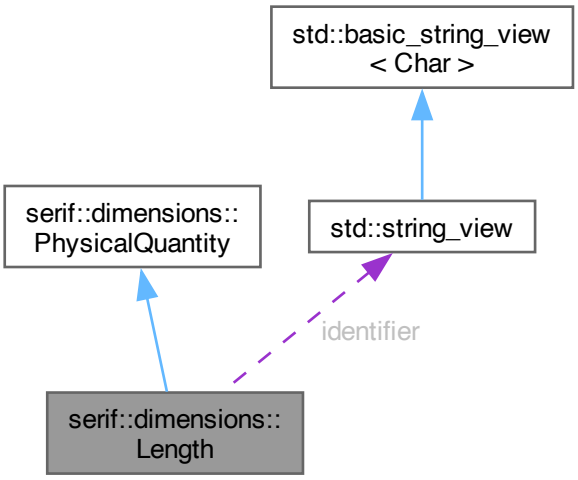
Dimension of Length.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Length:



Collaboration diagram for serif::dimensions::Length:



Static Public Attributes

- static constexpr std::string_view identifier = ”length”

10.299.1 Detailed Description

Dimension of Length.

10.299.2 Member Data Documentation

10.299.2.1 identifier

std::string_view serif::dimensions::Length::identifier = ”length” [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.300 serif::geometry::deformation::keys::Linearization Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

10.301 serif::geometry::surface::keys::Linearization Struct Reference

#include <samples.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/samples.hpp

10.302 serif::operators::backend::mfem::nodal_keys::Linearization Struct Reference

#include <nodal_inputs.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/nodal_inputs.hpp

10.303 serif::operators::backend::mfem::volume_keys::Linearization Struct Reference

#include <volume_data.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.304 serif::geometry::deformation::backend::mfem::detail::LocatedBoundaryPoint Struct Reference

Keep the best inverse boundary-map result and its physical residual.

#include <radial_locator.hpp>

Public Attributes

- int boundary_element
- ::mfem::IntegrationPoint point

10.304.1 Detailed Description

Keep the best inverse boundary-map result and its physical residual.

10.304.2 Member Data Documentation

10.304.2.1 boundary_element

int serif::geometry::deformation::backend::mfem::detail::LocatedBoundaryPoint::boundary_element

10.304.2.2 point

::mfem::IntegrationPoint serif::geometry::deformation::backend::mfem::detail::LocatedBoundaryPoint::point

The documentation for this struct was generated from the following file:

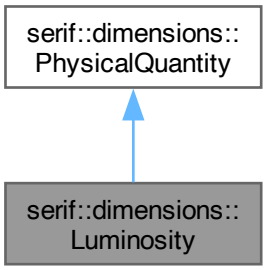
- src/include/serif/geometry/deformation/backend/mfem/detail/radial_locator.hpp

10.305 serif::dimensions::Luminosity Struct Reference

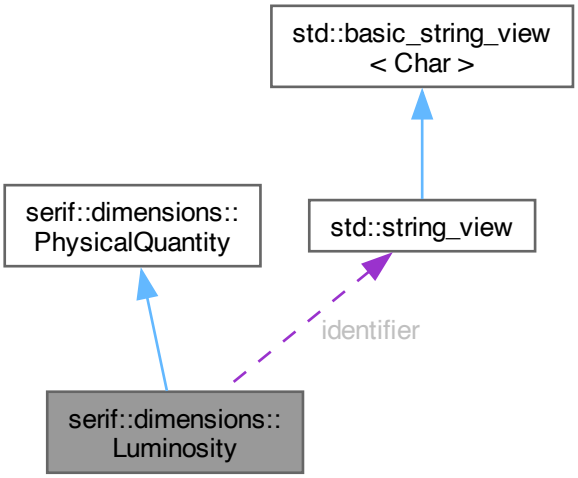
Dimension of Luminosity.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Luminosity:



Collaboration diagram for serif::dimensions::Luminosity:



Static Public Attributes

- static constexpr std::string_view identifier = ”luminosity”

10.305.1 Detailed Description

Dimension of Luminosity.

10.305.2 Member Data Documentation

10.305.2.1 identifier

std::string_view serif::dimensions::Luminosity::identifier = ”luminosity” [static], [constexpr]

The documentation for this struct was generated from the following file:

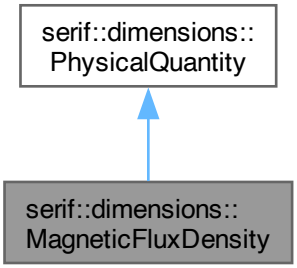
- src/include/serif/dimensions/dimensions.hpp

10.306 serif::dimensions::MagneticFluxDensity Struct Reference

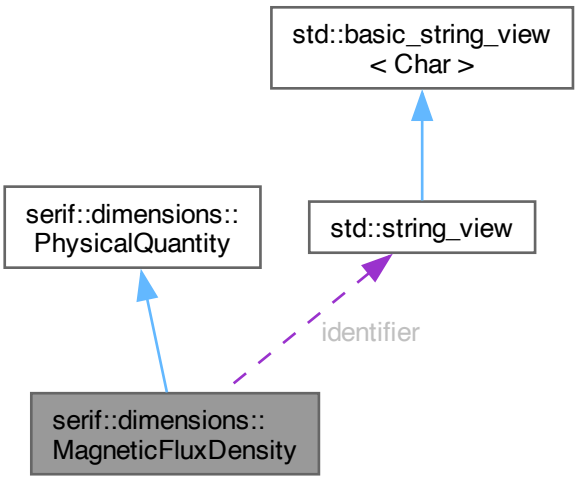
Dimension of Magnetic Flux Density.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::MagneticFluxDensity:



Collaboration diagram for serif::dimensions::MagneticFluxDensity:



Static Public Attributes

- static constexpr std::string_view identifier = "magnetic_flux_density"

10.306.1 Detailed Description

Dimension of Magnetic Flux Density.

10.306.2 Member Data Documentation

10.306.2.1 identifier

std::string_view serif::dimensions::MagneticFluxDensity::identifier = "magnetic_flux_density" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.307 serif::operators::backend::mfem::MappingCoefficients Struct Reference

#include <term_coefficients.hpp>

Public Attributes

- const ::mfem::FiniteElementSpace & binding
- const ::mfem::Vector & values

10.307.1 Member Data Documentation

10.307.1.1 binding

const ::mfem::FiniteElementSpace& serif::operators::backend::mfem::MappingCoefficients::binding

10.307.1.2 values

const ::mfem::Vector& serif::operators::backend::mfem::MappingCoefficients::values

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/term_coefficients.hpp

10.308 serif::mapping::MappingPointContext Struct Reference

#include <context.hpp>

Public Attributes

- mfem::Vector reference_position
- mfem::Vector displaced_position
- mfem::Vector physical_position
- mfem::DenseMatrix displacement_jacobian
- mfem::DenseMatrix mapping_jacobian
- mfem::DenseMatrix inverse_mapping_jacobian
- double mapping_determinate {0.0}
- bool compactified {false}

10.308.1 Member Data Documentation

10.308.1.1 compactified

bool serif::mapping::MappingPointContext::compactified {false}

10.308.1.2 displaced_position

mfem::Vector serif::mapping::MappingPointContext::displaced_position

10.308.1.3 displacement_jacobian

mfem::DenseMatrix serif::mapping::MappingPointContext::displacement_jacobian

10.308.1.4 inverse_mapping_jacobian

mfem::DenseMatrix serif::mapping::MappingPointContext::inverse_mapping_jacobian

10.308.1.5 mapping_determinate

double serif::mapping::MappingPointContext::mapping_determinate {0.0}

10.308.1.6 mapping_jacobian

mfem::DenseMatrix serif::mapping::MappingPointContext::mapping_jacobian

10.308.1.7 physical_position

mfem::Vector serif::mapping::MappingPointContext::physical_position

10.308.1.8 reference_position

mfem::Vector serif::mapping::MappingPointContext::reference_position

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/context.hpp

10.309 serif::mapping::MappingPointVariation Struct Reference

#include <variation.hpp>

Public Attributes

- mfem::Vector displacement_variation
- mfem::Vector physical_position_variation
- mfem::DenseMatrix displacement_jacobian_variation
- mfem::DenseMatrix mapping_jacobian_variation
- mfem::DenseMatrix inverse_mapping_jacobian_variation
- double mapping_determinate_variation {0.0}

10.309.1 Member Data Documentation

10.309.1.1 displacement_jacobian_variation

mfem::DenseMatrix serif::mapping::MappingPointVariation::displacement_jacobian_variation

10.309.1.2 displacement_variation

mfem::Vector serif::mapping::MappingPointVariation::displacement_variation

10.309.1.3 inverse_mapping_jacobian_variation

mfem::DenseMatrix serif::mapping::MappingPointVariation::inverse_mapping_jacobian_variation

10.309.1.4 mapping_determinate_variation

double serif::mapping::MappingPointVariation::mapping_determinate_variation {0.0}

10.309.1.5 mapping_jacobian_variation

mfem::DenseMatrix serif::mapping::MappingPointVariation::mapping_jacobian_variation

10.309.1.6 physical_position_variation

mfem::Vector serif::mapping::MappingPointVariation::physical_position_variation

The documentation for this struct was generated from the following file:

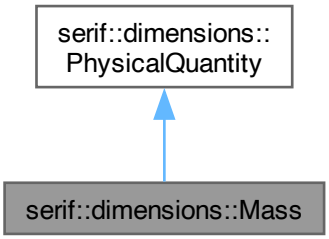
- src/include/serif/mapping/variation.hpp

10.310 serif::dimensions::Mass Struct Reference

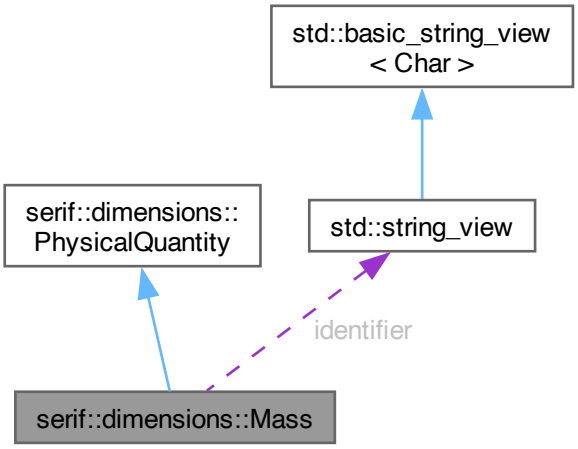
Dimension of mass.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Mass:



Collaboration diagram for serif::dimensions::Mass:



Static Public Attributes

- static constexpr std::string_view identifier = ”mass”

10.310.1 Detailed Description

Dimension of mass.

10.310.2 Member Data Documentation

10.310.2.1 identifier

std::string_view serif::dimensions::Mass::identifier = ”mass” [static], [constexpr]

The documentation for this struct was generated from the following file:

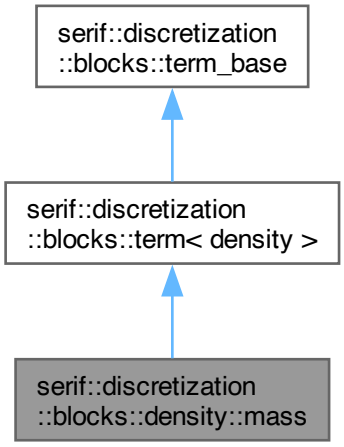
- src/include/serif/dimensions/dimensions.hpp

10.311 serif::discretization::blocks::density::mass Struct Reference

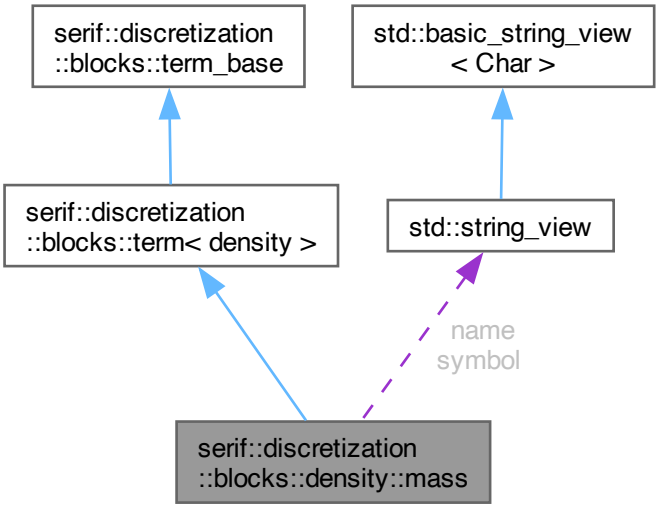
The mass density term for the system. Note that the naming here could be a bit confusing as FEM also uses the term ”mass” to represent the canonical mass matrix. This is a more standard physics reading of what is going on here. We define that this space is a second order scalar L2 space. This allows for the possibility of discontinuous density fields.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::density::mass:



Collaboration diagram for serif::discretization::blocks::density::mass:



Classes

- struct residual

  The residual block is a block that contains the residual of the mass density term. It is a scalar field that is associated with the mass density of the system.
- struct value

  The value block is a block that contains the value of the mass density term. It is a scalar field that is associated with the mass density of the system.

Public Types

- using FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::L2, 2, 0>

  second order scalar L2 space. Read the template parameters as follows: 1) L2 space, 2) second order, 3) scalar field (rank 0)
- using Support = domain::StellarDomains

  The support of the mass density term is the stellar domains since we do not allow material to exist in the vacuum domain.
- using Quantity = dimensions::MassDensity

  The quantity of the mass density term is the mass density.

Public Types inherited from serif::discretization::blocks::term< density >

- using field_type

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<mass>()
- static constexpr std::string_view symbol = " "

  Compile time symbol used for the term. Note that this is a UTF-8 character and may not render correctly in all environments. (TODO: Should we introduce an ASCII fallback?, something like make this a tuple of form <main, fallback>?)

10.311.1 Detailed Description

The mass density term for the system. Note that the naming here could be a bit confusing as FEM also uses the term "mass" to represent the canonical mass matrix. This is a more standard physics reading of what is going on here. We define that this space is a second order scalar L2 space. This allows for the possibility of discontinuous density fields.

10.311.2 Member Typedef Documentation

10.311.2.1 FunctionalSpace

using serif::discretization::blocks::density::mass::FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::L2, 2, 0>

second order scalar L2 space. Read the template parameters as follows: 1) L2 space, 2) second order, 3) scalar field (rank 0)

10.311.2.2 Quantity

using serif::discretization::blocks::density::mass::Quantity = dimensions::MassDensity

The quantity of the mass density term is the mass density.

10.311.2.3 Support

using serif::discretization::blocks::density::mass::Support = domain::StellarDomains

The support of the mass density term is the stellar domains since we do not allow material to exist in the vacuum domain.

10.311.3 Member Data Documentation

10.311.3.1 name

std::string_view serif::discretization::blocks::density::mass::name = meta::type_name<mass>() [static], [constexpr]

10.311.3.2 symbol

std::string_view serif::discretization::blocks::density::mass::symbol = " " [static], [constexpr]

Compile time symbol used for the term. Note that this is a UTF-8 character and may not render correctly in all environments. (TODO: Should we introduce an ASCII fallback?, something like make this a tuple of form <main, fallback>?)

The documentation for this struct was generated from the following file:

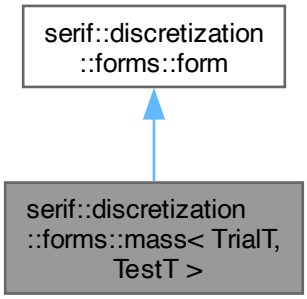
- src/include/serif/discretization/blocks/fields.hpp

10.312 serif::discretization::forms::mass< TrialT, TestT > Struct Template Reference

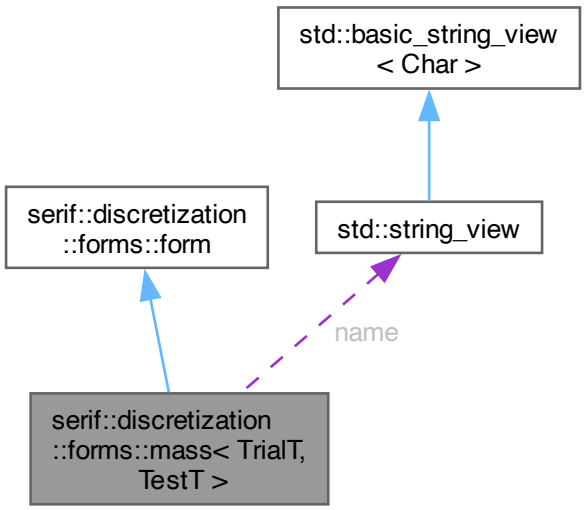
The mass form represents the mass matrix operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the mass operation.

#include <forms.hpp>

Inheritance diagram for serif::discretization::forms::mass< TrialT, TestT >:



Collaboration diagram for serif::discretization::forms::mass< TrialT, TestT >:



Public Types

- using trial_type = TrialT
  The trial term type for the mass form.
- using test_type = TestT
  The test term type for the mass form.
- using operand_types = utils::types::TypeList<Operand<TrialT>, Operand<TestT>>
  Type list of operand types for the mass form, used for compile-time reflection and iteration over the operands.

Static Public Attributes

- static constexpr std::string_view name = meta::template_name<mass>()
  Compile time reflected name of the form.

10.312.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::mass< TrialT, TestT >

The mass form represents the mass matrix operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the mass operation.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

It is important to not that we do not use mass here in the physics sense of the term; rather, mass is a common term in finite-element analysis. The mass matrix is a particular matrix that arrises in the weak discretization of a wide variety of PDEs. Consider the following example:

You have some scalar field $u(x,t)$ on some domain Ω which satisfies

$$\frac{\partial u}{\partial t} = f(x,t)$$

In the simplest case let every spatial point be independent such that there are no spatial derivatives. We first take this into weak form

$$\int_\Omega v \frac{\partial u}{\partial t} d\Omega = \int_\Omega v f(x,t) d\Omega$$

Now we must discretize this over some meshing, lets call the approximated solution u_h such that, approximating with N basis functions $\phi_j(x)$ we have

$$u \approx u_h = \sum_{j=1}^{N} c_j(t)\phi_j(x)$$

Then we can rewrite our weak form as

$$\int_\Omega v \sum_{j=1}^{N} c_j{}'(t)\phi_j(x) d\Omega = \int_\Omega v f(x,t) d\Omega$$

If we use the galerkin method such that our test function is the same as our basis functions, then we can write this as

$$\int_\Omega \phi_i(x) \sum_{j=1}^{N} c_j{}'(t)\phi_j(x) d\Omega = \int_\Omega \phi_i(x) f(x,t) d\Omega$$

Since we have defined the problem so that the basis functions are not time-dependent we can rearrange this as

$$\sum_j \left(\int_\Omega \phi_i \phi_j d\Omega \right) c_j{}'(t) = \int_\Omega \phi_i f(x,t) d\Omega$$

When then define

$$M_{ij} = \int_{\Omega} \phi_i \phi_j d\Omega$$

This is the so called mass matrix''. The careful reader will note that this really need have nothing to do with mass. Formally the mass matrix just represents the \f$L^{2}\f$ inner product of the basis functions. Historically (from some rudimentary googling) this seems to be a holdover from the early days of finite element analysis and models of the longitudinal vibrations of a elastic bar. You can go through that PDE following the same general steps as above and you also arrive at some \f$L^{2}\f$ inner product of the basis functions (along with another term). The inner product term turns out to be associated with the mass of the bar while the other term turns out to be associated with the stiffness of the bar (and therefore matrices of that form have become known asstiffness matrices'').

There is likely a good argument for us to use a different name, clarity may be improved. However, most resource on finite element analysis will use the term mass matrix so I have decided to stick with that for now.

10.312.2 Member Typedef Documentation

10.312.2.1 operand_types

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::mass< TrialT, TestT >::operand_types = utils::types::TypeList<Operand<TrialT>, Operand<TestT>>

Type list of operand types for the mass form, used for compile-time reflection and iteration over the operands.

10.312.2.2 test_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::mass< TrialT, TestT >::test_type = TestT

The test term type for the mass form.

10.312.2.3 trial_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::mass< TrialT, TestT >::trial_type = TrialT

The trial term type for the mass form.

10.312.3 Member Data Documentation

10.312.3.1 name

template<blocks::Term TrialT, blocks::Term TestT>
std::string_view serif::discretization::forms::mass< TrialT, TestT >::name = meta::template_name<mass>() [static], [constexpr]

Compile time reflected name of the form.

The documentation for this struct was generated from the following file:

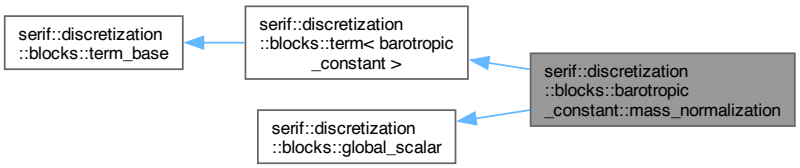
- src/include/serif/discretization/forms/forms.hpp

10.313 serif::discretization::blocks::barotropic_constant::mass_↩ normalization Struct Reference

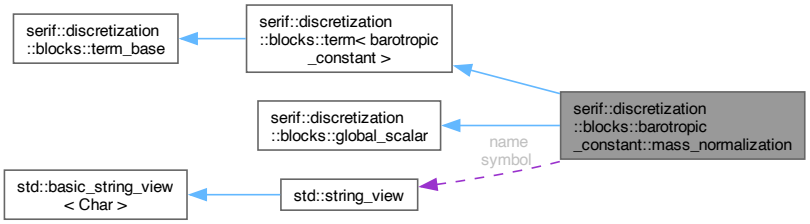
the mass_normalization term represents the mass normalization condition of the system. That is that it enforces that the total mass of the system is equal to the target mass in a weak sense. This is both a term and a global scalar.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::barotropic_constant::mass_normalization:



Collaboration diagram for serif::discretization::blocks::barotropic_constant::mass_normalization:



Public Types

- using Support = domain::StellarDomains

  The support of the mass normalization term is the stellar domains since we only want to enforce the mass normalization condition in the stellar domains.
- using Quantity = dimensions::SpecificEnthalpy

  As we are in teh barotropic case the mass normalization term is associated with the specific enthalpy. This will nto generalize to the non-barotropic case.

Public Types inherited from serif::discretization::blocks::term< barotropic_constant >

- using field_type

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<mass_normalization>()
  Compile time reflected name of the term.
- static constexpr std::string_view symbol = "C"
  Compile time symbol used for the term.

10.313.1 Detailed Description

the mass_normalization term represents the mass normalization condition of the system. That is that it enforces that the total mass of the system is equal to the target mass in a weak sense. This is both a term and a global scalar.

10.313.2 Member Typedef Documentation

10.313.2.1 Quantity

using serif::discretization::blocks::barotropic_constant::mass_normalization::Quantity = dimensions::SpecificEnthalpy

As we are in teh barotropic case the mass normalization term is associated with the specific enthalpy. This will nto generalize to the non-barotropic case.

10.313.2.2 Support

using serif::discretization::blocks::barotropic_constant::mass_normalization::Support = domain::StellarDomains

The support of the mass normalization term is the stellar domains since we only want to enforce the mass normalization condition in the stellar domains.

10.313.3 Member Data Documentation

10.313.3.1 name

std::string_view serif::discretization::blocks::barotropic_constant::mass_normalization::name = meta::type_name<mass_normalization>() [static], [constexpr]

Compile time reflected name of the term.

10.313.3.2 symbol

std::string_view serif::discretization::blocks::barotropic_constant::mass_normalization::symbol = "C" [static], [constexpr]

Compile time symbol used for the term.

The documentation for this struct was generated from the following file:

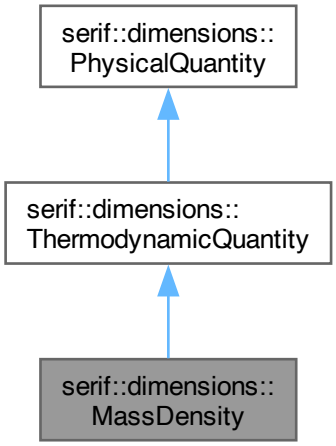
- src/include/serif/discretization/blocks/fields.hpp

10.314 serif::dimensions::MassDensity Struct Reference

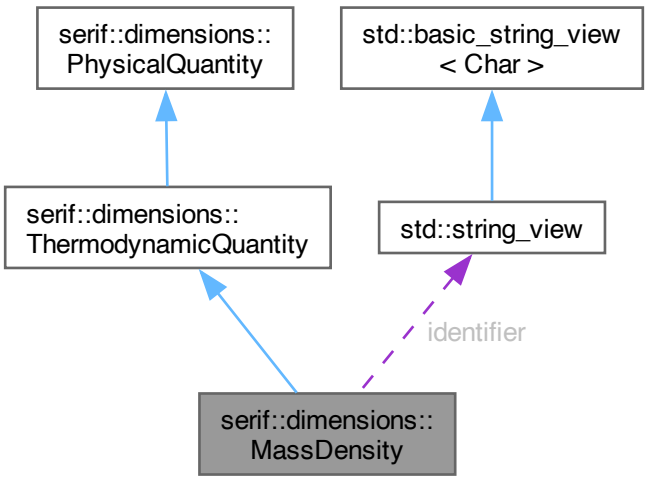
Dimension of Mass Density.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::MassDensity:



Collaboration diagram for serif::dimensions::MassDensity:



Static Public Attributes

- static constexpr std::string_view identifier = ”density”

10.314.1 Detailed Description

Dimension of Mass Density.

10.314.2 Member Data Documentation

10.314.2.1 identifier

std::string_view serif::dimensions::MassDensity::identifier = ”density” [static], [constexpr]

The documentation for this struct was generated from the following file:

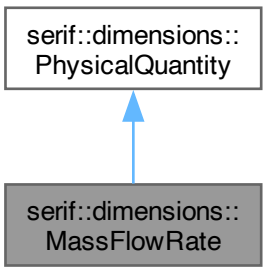
- src/include/serif/dimensions/dimensions.hpp

10.315 serif::dimensions::MassFlowRate Struct Reference

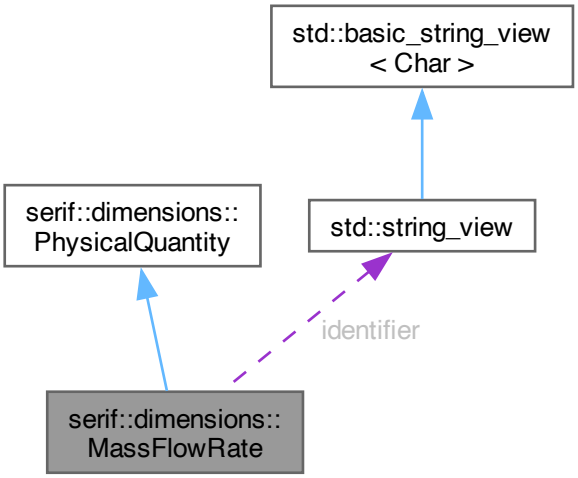
Dimension of Mass Flow Rate.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::MassFlowRate:



Collaboration diagram for serif::dimensions::MassFlowRate:



Static Public Attributes

- static constexpr std::string_view identifier = ”mass_flow_rate”

10.315.1 Detailed Description

Dimension of Mass Flow Rate.

10.315.2 Member Data Documentation

10.315.2.1 identifier

std::string_view serif::dimensions::MassFlowRate::identifier = ”mass_flow_rate” [static], [constexpr]

The documentation for this struct was generated from the following file:

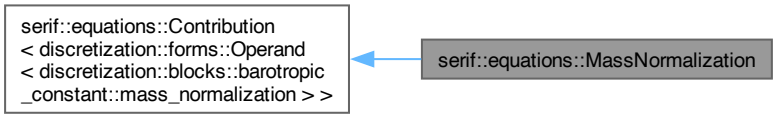
- src/include/serif/dimensions/dimensions.hpp

10.316 serif::equations::MassNormalization Class Reference

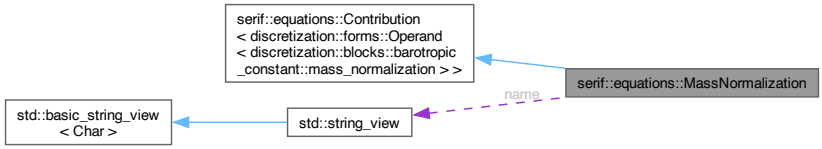
Close the unknown Bernoulli constant with integral_star rho dV - target_mass = 0.

#include <mass_normalization.hpp>

Inheritance diagram for serif::equations::MassNormalization:



Collaboration diagram for serif::equations::MassNormalization:



Public Types

- using Density = discretization::blocks::density::mass
- using DensityValue = discretization::forms::Operand<Density>
- using Operands = utils::types::TypeList<DensityValue>

Public Types inherited from serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::barotropic_constan

- using Test
- using TestOperand
- using Support

Public Member Functions

- MassNormalization (double target_mass)
- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const
- int additional_order (DensityValue, int) const
- double residual_offset () const

  Applied once after integration, including when this equation is scaled or subtracted.

Static Public Attributes

- static constexpr std::string_view name = "mass normalization"

Private Attributes

- double m_target_mass

10.316.1 Detailed Description

Close the unknown Bernoulli constant with integral_star rho dV - target_mass = 0.

10.316.2 Member Typedef Documentation

10.316.2.1 Density

using serif::equations::MassNormalization::Density = discretization::blocks::density::mass

10.316.2.2 DensityValue

using serif::equations::MassNormalization::DensityValue = discretization::forms::Operand<Density>

10.316.2.3 Operands

using serif::equations::MassNormalization::Operands = utils::types::TypeList<DensityValue>

10.316.3 Constructor & Destructor Documentation

10.316.3.1 MassNormalization()

serif::equations::MassNormalization::MassNormalization (
double target_mass) [inline], [explicit]

10.316.4 Member Function Documentation

10.316.4.1 additional_order()

int serif::equations::MassNormalization::additional_order (
DensityValue ,
int) const [inline], [nodiscard]

10.316.4.2 evaluate()

PointLinearization< Operands > serif::equations::MassNormalization::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

10.316.4.3 residual_offset()

double serif::equations::MassNormalization::residual_offset () const [inline], [nodiscard]

Applied once after integration, including when this equation is scaled or subtracted.

10.316.5 Member Data Documentation

10.316.5.1 m_target_mass

double serif::equations::MassNormalization::m_target_mass [private]

10.316.5.2 name

std::string_view serif::equations::MassNormalization::name = "mass normalization" [static], [constexpr]

The documentation for this class was generated from the following file:

- src/include/serif/equations/mass_normalization.hpp

10.317 serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, Catalog > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/detail/relations.hpp

10.318 serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Relations... > > Struct Template Reference

#include <relations.hpp>

Public Types

- using Matches = decltype(std::tuple_cat(std::declval<std::conditional_t<matches_pressure_relation<EOS, Bindings, Carrier, Relations>(), std::tuple<Relations>, std::tuple<>>>()...))

Static Public Attributes

- static constexpr std::size_t count = std::tuple_size_v<Matches>

10.318.1 Member Typedef Documentation

10.318.1.1 Matches

template<typename EOS, typename Bindings, typename Carrier, typename... Relations>
using serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Relations... > >::Matches = decltype(std::tuple_cat(std::declval<std::conditional_t<matches_pressure_relation<EOS, Bindings, Carrier, Relations>(), std::tuple<Relations>, std::tuple<>>>()...))

10.318.2 Member Data Documentation

10.318.2.1 count

template<typename EOS, typename Bindings, typename Carrier, typename... Relations>
std::size_t serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Relations... > >::count = std::tuple_size_v<Matches> [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/detail/relations.hpp

10.319 serif::utils::dependencies::detail::MergeKeyLists< ResultT, Lists > Struct Template Reference

#include <type_lists.hpp>

Public Types

- using type = ResultT

10.319.1 Member Typedef Documentation

10.319.1.1 type

template<typename ResultT, typename... Lists>
using serif::utils::dependencies::detail::MergeKeyLists< ResultT, Lists >::type = ResultT

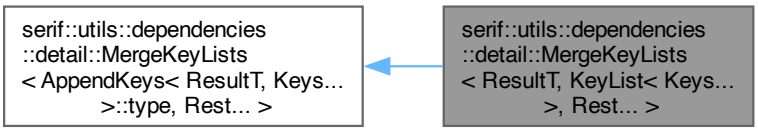
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/type_lists.hpp

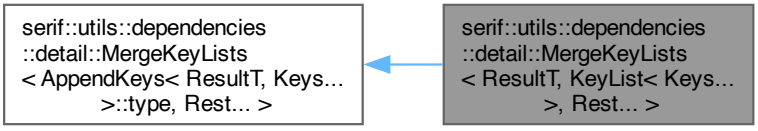
10.320 serif::utils::dependencies::detail::MergeKeyLists< ResultT, KeyList< Keys... >, Rest... > Struct Template Reference

#include <type_lists.hpp>

Inheritance diagram for serif::utils::dependencies::detail::MergeKeyLists< ResultT, KeyList< Keys... >, Rest... >:



Collaboration diagram for serif::utils::dependencies::detail::MergeKeyLists< ResultT, KeyList< Keys... >, Rest... >:



Public Types

- using type

Public Types inherited from serif::utils::dependencies::detail::MergeKeyLists< AppendKeys< ResultT, Keys... >::type, Rest... >

- using type

10.320.1 Member Typedef Documentation

10.320.1.1 type

using serif::utils::dependencies::detail::MergeKeyLists< ResultT, Lists >::type

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/type_lists.hpp

10.321 serif::operators::backend::mfem::MeshAndCoefficientBindings Struct Reference

#include <volume_inputs.hpp>

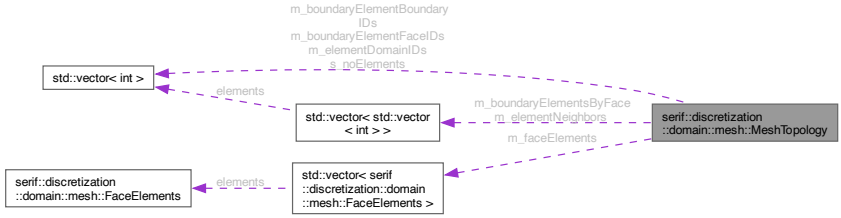
The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_inputs.hpp

10.322 serif::discretization::domain::mesh::MeshTopology Class Reference

#include <topology.hpp>

Collaboration diagram for serif::discretization::domain::mesh::MeshTopology:



Public Member Functions

- MeshTopology (const mfem::Mesh &mesh)
- int element_count () const noexcept
- int face_count () const noexcept
- int boundary_element_count () const noexcept
- int element_domain_id (int elementID) const
- int boundary_element_boundary_id (int boundaryElementID) const
- FaceElements face_elements (int faceID) const
- const std::vector< int > & element_neighbors (int elementID) const
- int boundary_element_face (int boundaryElementID) const
- const std::vector< int > & boundary_elements_on_face (int faceID) const

Private Attributes

- std::vector< int > m_elementDomainIDs
- std::vector< FaceElements > m_faceElements
- std::vector< std::vector< int > > m_elementNeighbors
- std::vector< int > m_boundaryElementBoundaryIDs
- std::vector< int > m_boundaryElementFaceIDs
- std::vector< std::vector< int > > m_boundaryElementsByFace

Static Private Attributes

- static const std::vector< int > s_noElements {}

10.322.1 Constructor & Destructor Documentation

10.322.1.1 MeshTopology()

serif::discretization::domain::mesh::MeshTopology::MeshTopology (
const mfem::Mesh & mesh) [explicit]

10.322.2 Member Function Documentation

10.322.2.1 boundary_element_boundary_id()

int serif::discretization::domain::mesh::MeshTopology::boundary_element_boundary_id (
int boundaryElementID) const [nodiscard]

10.322.2.2 boundary_element_count()

int serif::discretization::domain::mesh::MeshTopology::boundary_element_count () const [nodiscard], [noexcept]

10.322.2.3 boundary_element_face()

int serif::discretization::domain::mesh::MeshTopology::boundary_element_face (
int boundaryElementID) const [nodiscard]

10.322.2.4 boundary_elements_on_face()

const std::vector< int > & serif::discretization::domain::mesh::MeshTopology::boundary_elements_on_face (
int faceID) const [nodiscard]

10.322.2.5 element_count()

int serif::discretization::domain::mesh::MeshTopology::element_count () const [nodiscard], [noexcept]

10.322.2.6 element_domain_id()

int serif::discretization::domain::mesh::MeshTopology::element_domain_id (
int elementID) const [nodiscard]

10.322.2.7 element_neighbors()

const std::vector< int > & serif::discretization::domain::mesh::MeshTopology::element_neighbors (
int elementID) const [nodiscard]

10.322.2.8 face_count()

int serif::discretization::domain::mesh::MeshTopology::face_count () const [nodiscard], [noexcept]

10.322.2.9 face_elements()

FaceElements serif::discretization::domain::mesh::MeshTopology::face_elements (
int faceID) const [nodiscard]

10.322.3 Member Data Documentation

10.322.3.1 m_boundaryElementBoundaryIDs

std::vector<int> serif::discretization::domain::mesh::MeshTopology::m_boundaryElementBoundaryIDs [private]

10.322.3.2 m_boundaryElementFaceIDs

std::vector<int> serif::discretization::domain::mesh::MeshTopology::m_boundaryElementFaceIDs [private]

10.322.3.3 m_boundaryElementsByFace

std::vector<std::vector<int> > serif::discretization::domain::mesh::MeshTopology::m_boundaryElementsByFace [private]

10.322.3.4 m_elementDomainIDs

std::vector<int> serif::discretization::domain::mesh::MeshTopology::m_elementDomainIDs [private]

10.322.3.5 m_elementNeighbors

std::vector<std::vector<int> > serif::discretization::domain::mesh::MeshTopology::m_elementNeighbors [private]

10.322.3.6 m_faceElements

std::vector<FaceElements> serif::discretization::domain::mesh::MeshTopology::m_faceElements [private]

10.322.3.7 s_noElements

const std::vector< int > serif::discretization::domain::mesh::MeshTopology::s_noElements {} [static], [private]

The documentation for this class was generated from the following files:

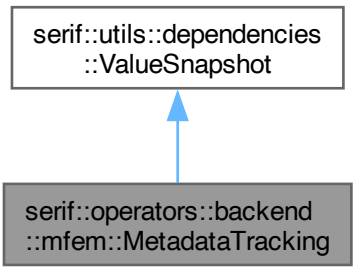
- src/include/serif/discretization/domain/mesh/topology.hpp
- src/lib/serif/discretization/domain/mesh/topology.cpp

10.323 serif::operators::backend::mfem::MetadataTracking Struct Reference

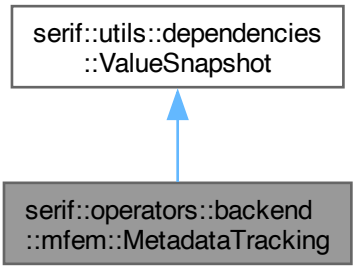
Metadata is a value; moving its temporary wrapper does not change the mesh or bindings it describes.

#include <volume_inputs.hpp>

Inheritance diagram for serif::operators::backend::mfem::MetadataTracking:



Collaboration diagram for serif::operators::backend::mfem::MetadataTracking:



Public Member Functions

- template<typename T>
  const void * identity (const T &) const

  Equality of the captured metadata includes every relevant identity and sequence number.

Public Member Functions inherited from serif::utils::dependencies::ValueSnapshot

- template<typename T>
  requires std::copy_constructible<T> && std::equality_comparable<T>
  T capture (const T &value) const

  Keep the accepted value independent of subsequent edits to a borrowed source.
- template<typename T>
  requires std::equality_comparable<T>
  bool matches (const T &captured, const T &value) const

  Compare content rather than a hash that could silently miss a change.

10.323.1 Detailed Description

Metadata is a value; moving its temporary wrapper does not change the mesh or bindings it describes.

10.323.2 Member Function Documentation

10.323.2.1 identity()

template<typename T>
const void * serif::operators::backend::mfem::MetadataTracking::identity (
const T &) const [inline], [nodiscard]

Equality of the captured metadata includes every relevant identity and sequence number.

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_inputs.hpp

10.324 serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< OperandListT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/backend/mfem/mfem_static_order.hpp

10.325 serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< utils::types::TypeList< OperandTs... > > Struct Template Reference

#include <mfem_static_order.hpp>

Static Public Attributes

- static constexpr int value = (resolve_operand_order<OperandTs>() + ... + 0)

10.325.1 Member Data Documentation

10.325.1.1 value

template<forms::IsOperand... OperandTs>
int serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< utils::types::TypeList< OperandTs... > >::value = (resolve_operand_order<OperandTs>() + ... + 0) [static], [constexpr]

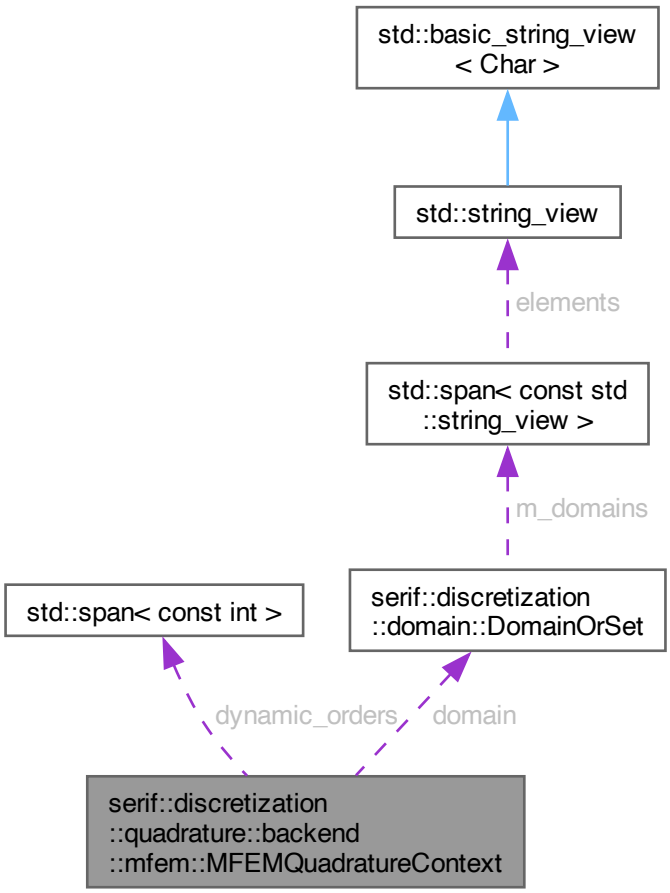
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/backend/mfem/mfem_static_order.hpp

10.326 serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext Struct Reference

#include <mfem_resolver.hpp>

Collaboration diagram for serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext:



Public Attributes

- QuadratureRole role = QuadratureRole::discretization
- ::mfem::Geometry::Type geometry = ::mfem::Geometry::INVALID
- int geometry_weight_order = 0
- std::span< const int > dynamic_orders {}
- domain::DomainOrSet domain = domain::AllDomains{}
- MappingKind mapping = MappingKind::none

10.326.1 Member Data Documentation

10.326.1.1 domain

domain::DomainOrSet serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext::domain = domain::AllDomains{}

10.326.1.2 dynamic_orders

std::span<const int> serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext::dynamic_orders {}

10.326.1.3 geometry

::mfem::Geometry::Type serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext::geometry = ::mfem::↩ Geometry::INVALID

10.326.1.4 geometry_weight_order

int serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext::geometry_weight_order = 0

10.326.1.5 mapping

MappingKind serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext::mapping = MappingKind::none

10.326.1.6 role

QuadratureRole serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext::role = QuadratureRole::discretization

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/backend/mfem/mfem_resolver.hpp

10.327 serif::discretization::quadrature::backend::mfem::MFEMResolver Class Reference

#include <mfem_resolver.hpp>

Collaboration diagram for serif::discretization::quadrature::backend::mfem::MFEMResolver:



Public Member Functions

- MFEMResolver (Policy policy) noexcept
- template<forms::Form FormT>
  MFEMRule resolve (const MFEMQuadratureContext &context) const
- template<forms::Form FormT, ConfigurableIntegrator IntegratorT>
  Resolution configure (IntegratorT &integrator, const MFEMQuadratureContext &context) const

Private Attributes

- Policy m_policy

10.327.1 Constructor & Destructor Documentation

10.327.1.1 MFEMResolver()

serif::discretization::quadrature::backend::mfem::MFEMResolver::MFEMResolver (
Policy policy) [inline], [explicit], [noexcept]

10.327.2 Member Function Documentation

10.327.2.1 configure()

template<forms::Form FormT, ConfigurableIntegrator IntegratorT>
Resolution serif::discretization::quadrature::backend::mfem::MFEMResolver::configure (
IntegratorT & integrator,
const MFEMQuadratureContext & context) const [inline]

10.327.2.2 resolve()

template<forms::Form FormT>
MFEMRule serif::discretization::quadrature::backend::mfem::MFEMResolver::resolve (
const MFEMQuadratureContext & context) const [inline], [nodiscard]

10.327.3 Member Data Documentation

10.327.3.1 m_policy

Policy serif::discretization::quadrature::backend::mfem::MFEMResolver::m_policy [private]

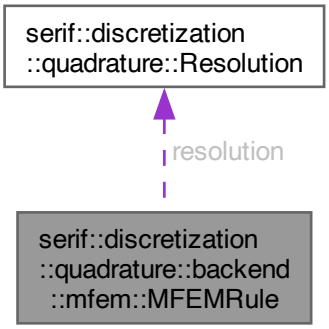
The documentation for this class was generated from the following file:

- src/include/serif/discretization/quadrature/backend/mfem/mfem_resolver.hpp

10.328 serif::discretization::quadrature::backend::mfem::MFEMRule Struct Reference

#include <mfem_resolver.hpp>

Collaboration diagram for serif::discretization::quadrature::backend::mfem::MFEMRule:



Public Attributes

- Resolution resolution {}
- const ::mfem::IntegrationRule * integration_rule = nullptr

10.328.1 Member Data Documentation

10.328.1.1 integration_rule

const ::mfem::IntegrationRule* serif::discretization::quadrature::backend::mfem::MFEMRule::integration_rule = nullptr

10.328.1.2 resolution

Resolution serif::discretization::quadrature::backend::mfem::MFEMRule::resolution {}

The documentation for this struct was generated from the following file:

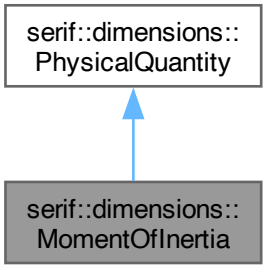
- src/include/serif/discretization/quadrature/backend/mfem/mfem_resolver.hpp

10.329 serif::dimensions::MomentOfInertia Struct Reference

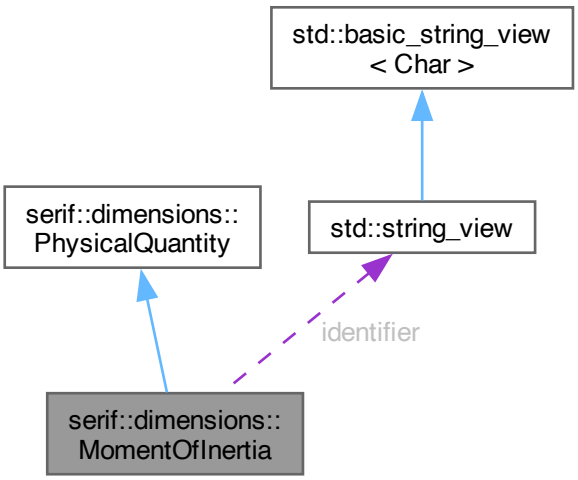
Dimension of Moment Of Inertia.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::MomentOfInertia:



Collaboration diagram for serif::dimensions::MomentOfInertia:



Static Public Attributes

- static constexpr std::string_view identifier = "moment_of_inertia"

10.329.1 Detailed Description

Dimension of Moment Of Inertia.

10.329.2 Member Data Documentation

10.329.2.1 identifier

std::string_view serif::dimensions::MomentOfInertia::identifier = "moment_of_inertia" [static], [constexpr]

The documentation for this struct was generated from the following file:

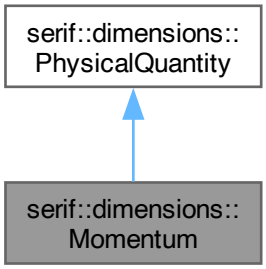
- src/include/serif/dimensions/dimensions.hpp

10.330 serif::dimensions::Momentum Struct Reference

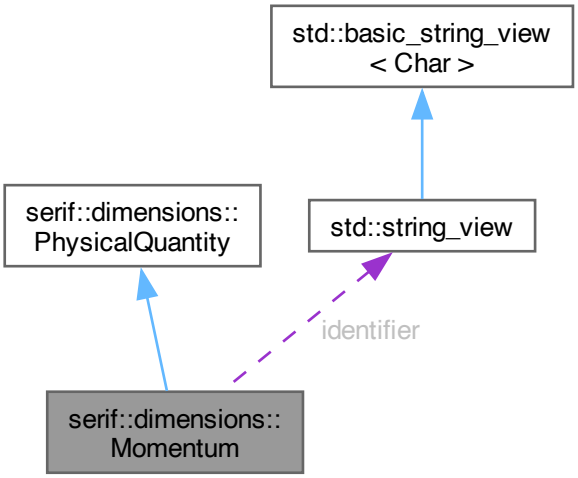
Dimension of Momentum.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Momentum:



Collaboration diagram for serif::dimensions::Momentum:



Static Public Attributes

- static constexpr std::string_view identifier = ”momentum”

10.330.1 Detailed Description

Dimension of Momentum.

10.330.2 Member Data Documentation

10.330.2.1 identifier

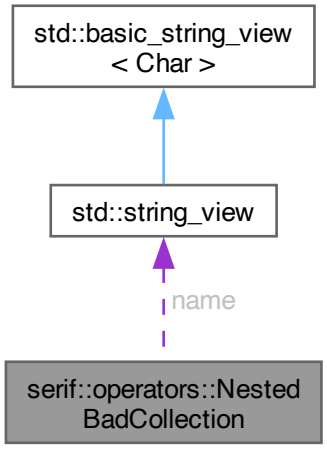
std::string_view serif::dimensions::Momentum::identifier = ”momentum” [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.331 serif::operators::NestedBadCollection Struct Reference

Collaboration diagram for serif::operators::NestedBadCollection:



Public Types

- using Test = Displacement

Public Member Functions

- auto contributions () const

Static Public Attributes

- static constexpr std::string_view name = "nested non-tuple contributions"

10.331.1 Member Typedef Documentation

10.331.1.1 Test

using serif::operators::NestedBadCollection::Test = Displacement

10.331.2 Member Function Documentation

10.331.2.1 contributions()

auto serif::operators::NestedBadCollection::contributions () const [inline]

10.331.3 Member Data Documentation

10.331.3.1 name

std::string_view serif::operators::NestedBadCollection::name = "nested non-tuple contributions" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/compile_time_checks/serif/operators/plan.cpp

10.332 serif::operators::backend::mfem::NodalData< EquationT, SchemaT > Struct Template Reference

A frozen nodal residual and derivative retain the selection needed to interpret every entry.

#include <nodal_data.hpp>

Public Types

- using Discretization = NodalDiscretization<EquationT, SchemaT>

Public Attributes

- Discretization discretization
- ::mfem::Vector residual
- ::mfem::DenseMatrix derivative

10.332.1 Detailed Description

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
struct serif::operators::backend::mfem::NodalData< EquationT, SchemaT >

A frozen nodal residual and derivative retain the selection needed to interpret every entry.

10.332.2 Member Typedef Documentation

10.332.2.1 Discretization

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
using serif::operators::backend::mfem::NodalData< EquationT, SchemaT >::Discretization = NodalDiscretization<EquationT, SchemaT>

10.332.3 Member Data Documentation

10.332.3.1 derivative

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
::mfem::DenseMatrix serif::operators::backend::mfem::NodalData< EquationT, SchemaT >::derivative

10.332.3.2 discretization

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
Discretization serif::operators::backend::mfem::NodalData< EquationT, SchemaT >::discretization

10.332.3.3 residual

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
::mfem::Vector serif::operators::backend::mfem::NodalData< EquationT, SchemaT >::residual

The documentation for this struct was generated from the following file:

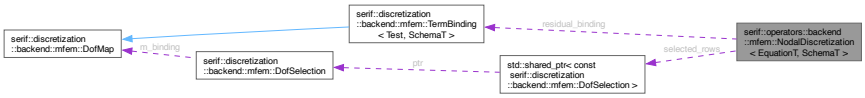
- src/include/serif/operators/backend/mfem/nodal_data.hpp

10.333 serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT > Struct Template Reference

Fixed coefficient ordering and selected rows shared by the graph and its accepted results.

#include <nodal_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >:



Public Types

- using Equation = EquationT
- using Test = typename EquationT::Test
- using ResidualBinding = discretization::backend::mfem::TermBinding<Test, SchemaT>
- using Value = discretization::forms::OperandValue<equations::TestOperandOf<EquationT>>

Public Attributes

- const ResidualBinding & residual_binding
- std::shared_ptr< const discretization::backend::mfem::DofSelection > selected_rows

Static Public Attributes

- static constexpr int components = discretization::forms::component_count<Value>

10.333.1 Detailed Description

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
struct serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >

Fixed coefficient ordering and selected rows shared by the graph and its accepted results.

10.333.2 Member Typedef Documentation

10.333.2.1 Equation

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
using serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >::Equation = EquationT

10.333.2.2 ResidualBinding

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
using serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >::ResidualBinding = discretization::backend::mfem::TermBindin
SchemaT>

10.333.2.3 Test

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
using serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >::Test = typename EquationT::Test

10.333.2.4 Value

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
using serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >::Value = discretization::forms::OperandValue<equations::TestC

10.333.3 Member Data Documentation

10.333.3.1 components

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
int serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >::components = discretization::forms::component_count<Value>
[static], [constexpr]

10.333.3.2 residual_binding

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
const ResidualBinding& serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >::residual_binding

10.333.3.3 selected_rows

template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
std::shared_ptr<const discretization::backend::mfem::DofSelection> serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >::selected_rows

The documentation for this struct was generated from the following file:

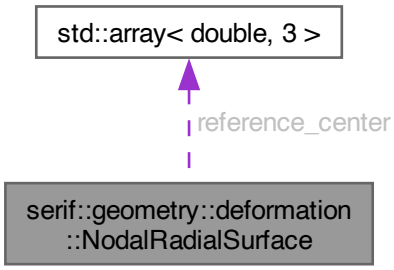
- src/include/serif/operators/backend/mfem/nodal_data.hpp

10.334 serif::geometry::deformation::NodalRadialSurface Struct Reference

Give each surface node an independent radial displacement, measured in physical length.

#include <models.hpp>

Collaboration diagram for serif::geometry::deformation::NodalRadialSurface:



Public Member Functions

- bool operator== (const NodalRadialSurface &) const =default
  Reference centers are physical inputs to dependency tracking.

Public Attributes

- std::array< double, 3 > reference_center {}

10.334.1 Detailed Description

Give each surface node an independent radial displacement, measured in physical length.

10.334.2 Member Function Documentation

10.334.2.1 operator==()

bool serif::geometry::deformation::NodalRadialSurface::operator== (
const NodalRadialSurface &) const [nodiscard], [default]

Reference centers are physical inputs to dependency tracking.

10.334.3 Member Data Documentation

10.334.3.1 reference_center

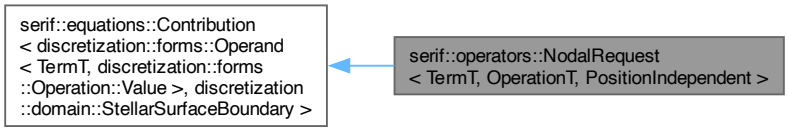
std::array<double, 3> serif::geometry::deformation::NodalRadialSurface::reference_center {}

The documentation for this struct was generated from the following file:

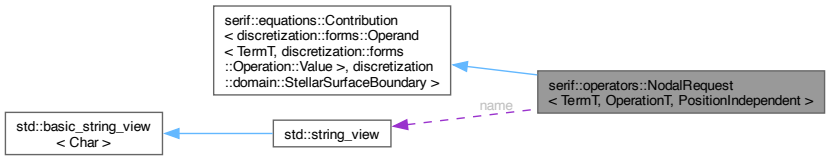
- src/include/serif/geometry/deformation/models.hpp

10.335 serif::operators::NodalRequest< TermT, OperationT, PositionIndependent > Struct Template Reference

Inheritance diagram for serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >:



Collaboration diagram for serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >:



Public Types

- using Operands = utils::types::TypeList<discretization::forms::Operand<TermT>>
- using Result = discretization::forms::OperandValue<discretization::forms::Operand<TermT, OperationT>>

Public Types inherited from serif::equations::Contribution< discretization::forms::Operand< TermT, discretization::forms::Operation::

- using Test
- using TestOperand
- using Support

Public Member Functions

- equations::PointLinearization< Operands, Result > evaluate (const equations::EquationPoint< Operands > &) const

Static Public Attributes

- static constexpr std::string_view name = "nodal capability test"
- static constexpr bool position_independent = PositionIndependent

10.335.1 Member Typedef Documentation

10.335.1.1 Operands

template<typename TermT, typename OperationT = discretization::forms::Operation::Value, bool PositionIndependent = true>
using serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >::Operands = utils::types::TypeList<discretization::forms::Operan

10.335.1.2 Result

template<typename TermT, typename OperationT = discretization::forms::Operation::Value, bool PositionIndependent = true>
using serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >::Result = discretization::forms::OperandValue<discretization::fo
OperationT>>

10.335.2 Member Function Documentation

10.335.2.1 evaluate()

template<typename TermT, typename OperationT = discretization::forms::Operation::Value, bool PositionIndependent = true>
equations::PointLinearization< Operands, Result > serif::operators::NodalRequest< TermT, OperationT, PositionIndependent
>::evaluate (
const equations::EquationPoint< Operands > &) const [inline]

10.335.3 Member Data Documentation

10.335.3.1 name

template<typename TermT, typename OperationT = discretization::forms::Operation::Value, bool PositionIndependent = true>
std::string_view serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >::name = "nodal capability test"
[static], [constexpr]

10.335.3.2 position_independent

template<typename TermT, typename OperationT = discretization::forms::Operation::Value, bool PositionIndependent = true>
bool serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >::position_independent = PositionIndependent
[static], [constexpr]

The documentation for this struct was generated from the following file:

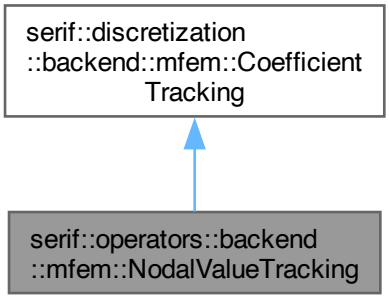
- src/compile_time_checks/serif/operators/plan.cpp

10.336 serif::operators::backend::mfem::NodalValueTracking Struct Reference

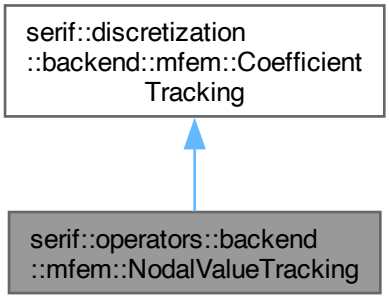
Moving the context or copying equal coefficients does not invalidate the nodal calculation.

#include <nodal_inputs.hpp>

Inheritance diagram for serif::operators::backend::mfem::NodalValueTracking:



Collaboration diagram for serif::operators::backend::mfem::NodalValueTracking:



Public Member Functions

- const void * identity (const SelectedNodalValues &input) const

  Selections with different interpretations must never share coefficient snapshots.

Public Member Functions inherited from serif::discretization::backend::mfem::CoefficientTracking

- template<typename ViewT>
  bool matches (const ::mfem::Vector &captured, const ViewT &view) const

  Check whether the current coefficients equal a captured snapshot.
- template<typename ViewT>
  ::mfem::Vector capture (const ViewT &view)

Additional Inherited Members

Static Public Member Functions inherited from serif::discretization::backend::mfem::CoefficientTracking

- template<typename ViewT>
  static ::mfem::Vector capture (const ViewT &view)

  Copy the current coefficient values into an independent snapshot.
- template<typename ViewT>
  static const void * identity (const ViewT &view)

  Identify the persistent binding associated with the coefficients.

10.336.1 Detailed Description

Moving the context or copying equal coefficients does not invalidate the nodal calculation.

10.336.2 Member Function Documentation

10.336.2.1 identity()

const void * serif::operators::backend::mfem::NodalValueTracking::identity (
const SelectedNodalValues & input) const [inline], [nodiscard]

Selections with different interpretations must never share coefficient snapshots.

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/nodal_inputs.hpp

10.337 serif::discretization::forms::Operation::NormalTrace Struct Reference

Operation tag for evaluating the normal flux along a boundary.

#include <operands.hpp>

10.337.1 Detailed Description

Operation tag for evaluating the normal flux along a boundary.

The documentation for this struct was generated from the following file:

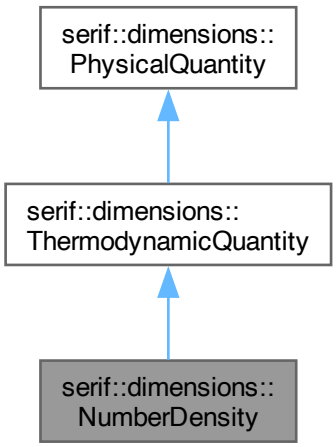
- src/include/serif/discretization/forms/operands.hpp

10.338 serif::dimensions::NumberDensity Struct Reference

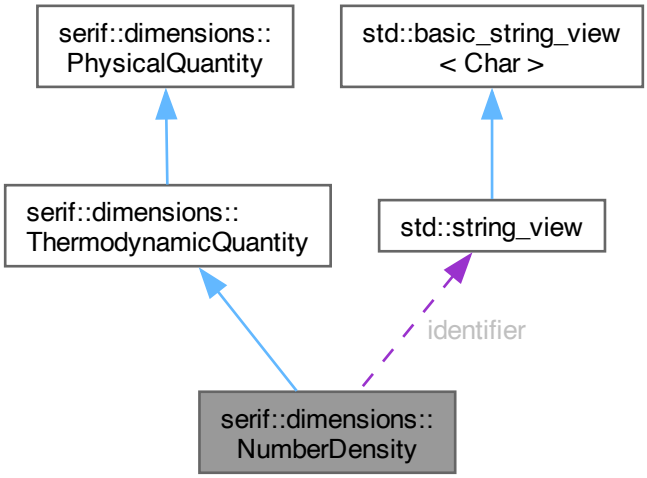
Dimension of Number Density.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::NumberDensity:



Collaboration diagram for serif::dimensions::NumberDensity:



Static Public Attributes

- static constexpr std::string_view identifier = "number_density"

10.338.1 Detailed Description

Dimension of Number Density.

10.338.2 Member Data Documentation

10.338.2.1 identifier

std::string_view serif::dimensions::NumberDensity::identifier = "number_density" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.339 serif::operators::backend::mfem::NumericalWork Struct Reference

Numerical work derived from the dependency engine's most recent report.

#include <preparation.hpp>

Public Member Functions

- bool did_work () const noexcept
  Report whether numerical preparation refreshed any samples or physics.

Public Attributes

- std::size_t reused_elements {0}
- std::size_t geometry_points {0}
- std::size_t sampled_operands {0}
- std::size_t physics_points {0}
- std::size_t refreshed_scalars {0}

10.339.1 Detailed Description

Numerical work derived from the dependency engine's most recent report.

10.339.2 Member Function Documentation

10.339.2.1 did_work()

bool serif::operators::backend::mfem::NumericalWork::did_work () const [inline], [nodiscard], [noexcept]

Report whether numerical preparation refreshed any samples or physics.

10.339.3 Member Data Documentation

10.339.3.1 geometry_points

std::size_t serif::operators::backend::mfem::NumericalWork::geometry_points {0}

10.339.3.2 physics_points

std::size_t serif::operators::backend::mfem::NumericalWork::physics_points {0}

10.339.3.3 refreshed_scalars

std::size_t serif::operators::backend::mfem::NumericalWork::refreshed_scalars {0}

10.339.3.4 reused_elements

std::size_t serif::operators::backend::mfem::NumericalWork::reused_elements {0}

10.339.3.5 sampled_operands

std::size_t serif::operators::backend::mfem::NumericalWork::sampled_operands {0}

The documentation for this struct was generated from the following file:

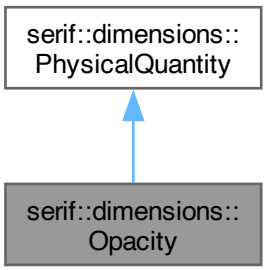
- src/include/serif/operators/backend/mfem/preparation.hpp

10.340 serif::dimensions::Opacity Struct Reference

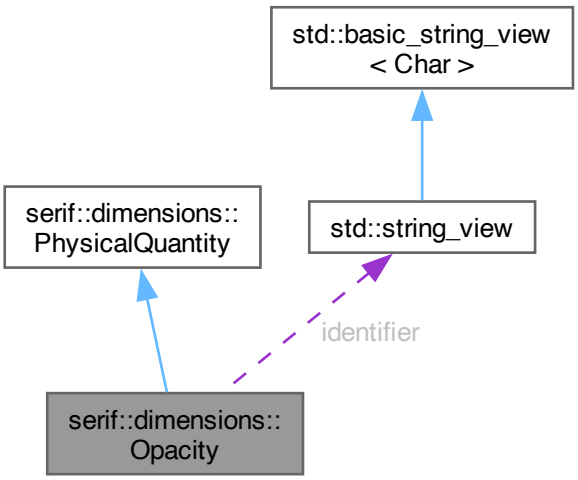
Dimension of Opacity.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Opacity:



Collaboration diagram for serif::dimensions::Opacity:



Static Public Attributes

- static constexpr std::string_view identifier = "opacity"

10.340.1 Detailed Description

Dimension of Opacity.

Note

we are not currently using this in the code since we are still using a polytropic eos. However, when we actually get to using this we will likely want to break this out into more specific and meaningful opacity dimensions (RMO, per unit wavelength, etc...)

10.340.2 Member Data Documentation

10.340.2.1 identifier

std::string_view serif::dimensions::Opacity::identifier = "opacity" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.341 serif::discretization::forms::Operand< TermT, OperationT > Struct Template Reference

Compile time representation of some operand. An operand associates a term to an operation type (e.g. the gradient of gravitational potential where gravitational potential is the term and gradient is the operation).

#include <operands.hpp>

Public Types

- using term_type = TermT
- using operation_type = OperationT

10.341.1 Detailed Description

template<blocks::Term TermT, IsOperation OperationT = Operation::Value>
struct serif::discretization::forms::Operand< TermT, OperationT >

Compile time representation of some operand. An operand associates a term to an operation type (e.g. the gradient of gravitational potential where gravitational potential is the term and gradient is the operation).

Template Parameters

TermT	Term Type (e.g. gravitational potential, mass density)
OperationT	The operation to associate to that term, must be one of the operations from Operation (enforced by satisfying IsOperation).

Note

An operand does not know how to do anything. Like a lot of the other parts of SERiF at this level this is compile time metadata which allows the compiler to select the correct overloads, without runtime overhead, in order to run the correct computations. Do not be fooled into thinking that this actually has any computations burried within it.

10.341.2 Member Typedef Documentation

10.341.2.1 operation_type

template<blocks::Term TermT, IsOperation OperationT = Operation::Value>
using serif::discretization::forms::Operand< TermT, OperationT >::operation_type = OperationT

10.341.2.2 term_type

template<blocks::Term TermT, IsOperation OperationT = Operation::Value>
using serif::discretization::forms::Operand< TermT, OperationT >::term_type = TermT

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/operands.hpp

10.342 serif::operators::backend::mfem::OperandEvaluator< OperandT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/operand_evaluation.hpp

10.343 serif::operators::backend::mfem::OperandEvaluator< OperandT > Struct Template Reference

Scalar values and H1/H(div) vector values or divergences share coefficient contractions.

#include <operand_evaluation.hpp>

Public Types

- using Term = typename OperandT::term_type
- using Value = discretization::forms::OperandValue<OperandT>
- using ReferenceValue = std::conditional_t<h1_divergence, std::array<double, 9>, Value>

Static Public Member Functions

- static void sample_basis (const discretization::backend::mfem::DofMap &binding, int element, const ::mfem::IntegrationRule &rule, BasisEvaluation &basis, TensorBasisCache &cache, ContractionMode mode, ::mfem::ElementTransformation &transformation)

  A global coefficient has basis one; spatial operands use their native element basis.
- static void sample_basis (const ::mfem::FiniteElement &element, const ::mfem::IntegrationRule &rule, BasisEvaluation &basis, TensorBasisCache &cache, ContractionMode mode, ::mfem::Element↩ Transformation &transformation)

  Sample a canonical tensor/dense basis, then apply MFEM's fixed-mesh mapping.
- static void map_basis_point (int point, const mapping::MappingPointContext &context, const BasisEvaluation &reference, BasisEvaluation &result, const mapping::MappingPointVariation *variation=nullptr)

  Apply SERiF's map to small tensor point maps, or to dense basis columns on other elements.

- static void evaluate (const BasisEvaluation &basis, const ::mfem::Vector &coefficients, ::mfem::Vector &values, BasisWorkspace &workspace)

  Apply a cached tensor or dense basis using reusable contraction scratch.
- static void integrate (const BasisEvaluation &basis, const ::mfem::Vector &values, ::mfem::Vector &result, BasisWorkspace &workspace)

  Transpose the same contraction; global scalar reductions retain compensated summation.
- static ReferenceValue reference_value (const ::mfem::Vector &values, int point)

  Read the reference data needed for one physical operand sample.
- static Value physical_value (const ReferenceValue &reference, const mapping::MappingPointContext &context)

  Apply only SERiF's additional fixed-mesh to physical map (MFEM's map is already included).
- static Value geometry_variation (const ReferenceValue &reference, const mapping::MappingPointContext &context, const mapping::MappingPointVariation &variation)

  At fixed coefficients: H1 divergence varies through A^-1; H(div) through A/j or 1/j.

Static Public Attributes

- static constexpr int components = discretization::forms::component_count<Value>
- static constexpr bool global_scalar = discretization::blocks::GlobalScalarTerm<Term>
- static constexpr bool hdiv
- static constexpr bool divergence = std::same_as<typename OperandT::operation_type, discretization::forms::Operatio
- static constexpr bool vector_copies
- static constexpr bool h1_divergence = vector_copies && divergence
- static constexpr bool geometry_dependent = hdiv || h1_divergence
- static constexpr int reference_components = h1_divergence ? 9 : components

Static Private Member Functions

- static double gradient_trace (const ReferenceValue &gradient, const ::mfem::DenseMatrix &inverse)

  div_x(u) = trace(G A^-1); replacing A^-1 by its variation gives delta(div_x(u)).

10.343.1 Detailed Description

template<VolumeOperand OperandT>
struct serif::operators::backend::mfem::OperandEvaluator< OperandT >

Scalar values and H1/H(div) vector values or divergences share coefficient contractions.

10.343.2 Member Typedef Documentation

10.343.2.1 ReferenceValue

template<VolumeOperand OperandT>
using serif::operators::backend::mfem::OperandEvaluator< OperandT >::ReferenceValue = std::conditional_t<h1_divergence, std::array<double, 9>, Value>

10.343.2.2 Term

template<VolumeOperand OperandT>
using serif::operators::backend::mfem::OperandEvaluator< OperandT >::Term = typename OperandT::term_type

10.343.2.3 Value

template<VolumeOperand OperandT>
using serif::operators::backend::mfem::OperandEvaluator< OperandT >::Value = discretization::forms::OperandValue<OperandT>

10.343.3 Member Function Documentation

10.343.3.1 evaluate()

template<VolumeOperand OperandT>
serif::operators::backend::mfem::OperandEvaluator< OperandT >::evaluate (
const BasisEvaluation & basis,
const ::mfem::Vector & coefficients,
::mfem::Vector & values,
BasisWorkspace & workspace) [inline], [static]

Apply a cached tensor or dense basis using reusable contraction scratch.

10.343.3.2 geometry_variation()

template<VolumeOperand OperandT>
static Value serif::operators::backend::mfem::OperandEvaluator< OperandT >::geometry_variation (
const ReferenceValue & reference,
const mapping::MappingPointContext & context,
const mapping::MappingPointVariation & variation) [inline], [static]

At fixed coefficients: H1 divergence varies through A^-1; H(div) through A/j or 1/j.

10.343.3.3 gradient_trace()

template<VolumeOperand OperandT>
static double serif::operators::backend::mfem::OperandEvaluator< OperandT >::gradient_trace (
const ReferenceValue & gradient,
const ::mfem::DenseMatrix & inverse) [inline], [static], [private]

div_x(u) = trace(G A^-1); replacing A^-1 by its variation gives delta(div_x(u)).

10.343.3.4 integrate()

```
template<VolumeOperand OperandT>
static void serif::operators::backend::mfem::OperandEvaluator< OperandT >::integrate (
        const BasisEvaluation & basis,
        const ::mfem::Vector & values,
        ::mfem::Vector & result,
        BasisWorkspace & workspace)   [inline], [static]
```

Transpose the same contraction; global scalar reductions retain compensated summation.

10.343.3.5 map_basis_point()

```
template<VolumeOperand OperandT>
static void serif::operators::backend::mfem::OperandEvaluator< OperandT >::map_basis_point (
        int point,
        const mapping::MappingPointContext & context,
        const BasisEvaluation & reference,
        BasisEvaluation & result,
        const mapping::MappingPointVariation * variation = nullptr)   [inline], [static]
```

Apply SERiF's map to small tensor point maps, or to dense basis columns on other elements.

10.343.3.6 physical_value()

```
template<VolumeOperand OperandT>
static Value serif::operators::backend::mfem::OperandEvaluator< OperandT >::physical_value (
        const ReferenceValue & reference,
        const mapping::MappingPointContext & context)   [inline], [static]
```

Apply only SERiF's additional fixed-mesh to physical map (MFEM's map is already included).

10.343.3.7 reference_value()

```
template<VolumeOperand OperandT>
static ReferenceValue serif::operators::backend::mfem::OperandEvaluator< OperandT >::reference_value (
        const ::mfem::Vector & values,
        int point)   [inline], [static]
```

Read the reference data needed for one physical operand sample.

10.343.3.8 sample_basis() [1/2]

```
template<VolumeOperand OperandT>
static void serif::operators::backend::mfem::OperandEvaluator< OperandT >::sample_basis (
        const ::mfem::FiniteElement & element,
        const ::mfem::IntegrationRule & rule,
        BasisEvaluation & basis,
        TensorBasisCache & cache,
        ContractionMode mode,
        ::mfem::ElementTransformation & transformation)   [inline], [static]
```

Sample a canonical tensor/dense basis, then apply MFEM's fixed-mesh mapping.

10.343.3.9 sample_basis() [2/2]

template<VolumeOperand OperandT>
static void serif::operators::backend::mfem::OperandEvaluator< OperandT >::sample_basis (
const discretization::backend::mfem::DofMap & binding,
int element,
const ::mfem::IntegrationRule & rule,
BasisEvaluation & basis,
TensorBasisCache & cache,
ContractionMode mode,
::mfem::ElementTransformation & transformation) [inline], [static]

A global coefficient has basis one; spatial operands use their native element basis.

10.343.4 Member Data Documentation

10.343.4.1 components

template<VolumeOperand OperandT>
int serif::operators::backend::mfem::OperandEvaluator< OperandT >::components = discretization::forms::component_count<Value>
[static], [constexpr]

10.343.4.2 divergence

template<VolumeOperand OperandT>
bool serif::operators::backend::mfem::OperandEvaluator< OperandT >::divergence = std::same_as<typename OperandT↩
::operation_type, discretization::forms::Operation::Divergence> [static], [constexpr]

10.343.4.3 geometry_dependent

template<VolumeOperand OperandT>
bool serif::operators::backend::mfem::OperandEvaluator< OperandT >::geometry_dependent = hdiv || h1_divergence [static],
[constexpr]

10.343.4.4 global_scalar

template<VolumeOperand OperandT>
bool serif::operators::backend::mfem::OperandEvaluator< OperandT >::global_scalar = discretization::blocks::GlobalScalar↩
Term<Term> [static], [constexpr]

10.343.4.5 h1_divergence

template<VolumeOperand OperandT>
bool serif::operators::backend::mfem::OperandEvaluator< OperandT >::h1_divergence = vector_copies && divergence
[static], [constexpr]

10.343.4.6 hdiv

template<VolumeOperand OperandT>
bool serif::operators::backend::mfem::OperandEvaluator< OperandT >::hdiv [static], [constexpr]

Initial value:

```
= [] {
        if constexpr (global_scalar) {
            return false;
        } else {
            return discretization::blocks::SpatialTermTraits<Term>::family == discretization::functional_space::Family::hdiv;
        }
    }()
```

10.343.4.7 reference_components

template<VolumeOperand OperandT>
int serif::operators::backend::mfem::OperandEvaluator< OperandT >::reference_components = h1_divergence ? 9 : components [static], [constexpr]

10.343.4.8 vector_copies

template<VolumeOperand OperandT>
bool serif::operators::backend::mfem::OperandEvaluator< OperandT >::vector_copies [static], [constexpr]

Initial value:

```
= [] {
        if constexpr (global_scalar) {
            return false;
        } else {
            return !hdiv && discretization::blocks::SpatialTermTraits<Term>::rank == 1;
        }
    }()
```

The documentation for this struct was generated from the following files:

- src/include/serif/operators/backend/mfem/operand_evaluation.hpp
- src/include/serif/operators/backend/mfem/element_preparation.hpp

10.344 serif::discretization::forms::OperandResult< OperandT > Struct Template Reference

General case for compile time metadata associating some operand to the value type of its result. These use the OperationValue struct and the information in the operand to determine the value type of the result of that operand.

10.344.1 Detailed Description

template<typename OperandT>
struct serif::discretization::forms::OperandResult< OperandT >

General case for compile time metadata associating some operand to the value type of its result. These use the OperationValue struct and the information in the operand to determine the value type of the result of that operand.

Template Parameters

OperandT	Operand Type

The documentation for this struct was generated from the following file:

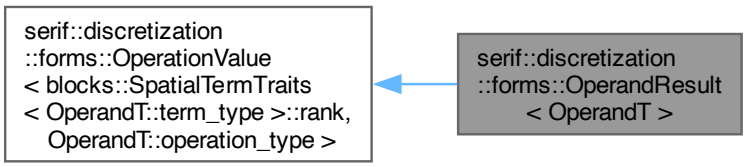
- src/include/serif/discretization/forms/value.hpp

10.345 serif::discretization::forms::OperandResult< OperandT > Struct Template Reference

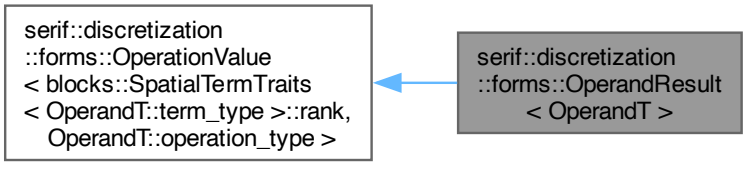
Specialization of the OperandResult struct for operands that are valid operands (satisfy the IsOperand concept) and are spatial terms (e.g defined over the domain, think mass density). This specialization uses the OperationValue struct to determine the value type of the result of the operand based on its term type and operation type.

#include <value.hpp>

Inheritance diagram for serif::discretization::forms::OperandResult< OperandT >:



Collaboration diagram for serif::discretization::forms::OperandResult< OperandT >:



Public Types

- using type = double

10.345.1 Detailed Description

template<IsOperand OperandT>
requires blocks::SpatialTerm<typename OperandT::term_type>
struct serif::discretization::forms::OperandResult< OperandT >

Specialization of the OperandResult struct for operands that are valid operands (satisfy the IsOperand concept) and are spatial terms (e.g defined over the domain, think mass density). This specialization uses the OperationValue struct to determine the value type of the result of the operand based on its term type and operation type.

Specialization of the OperandResult struct for operands that are valid operands (satisfy the IsOperand concept) and are global scalar terms (e.g one value over the domain, think total mass). This specialization defines that any operand that is a global scalar term will result in a double when evaluated.

Template Parameters

OperandT	The operand type to determine the result value type for.

Think of this a compile time way to say "this operand (which recall is an association between a form and a term) will result in a tensor of rank x when evaluated on a term of rank y".

Template Parameters

OperandT	The operand type to determine the result value type for.

10.345.2 Member Typedef Documentation

10.345.2.1 type

template<IsOperand OperandT>
using serif::discretization::forms::OperandResult< OperandT >::type = double

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/value.hpp

10.346 serif::operators::backend::mfem::OperandSamples Struct Reference

Reference samples remain available for derivatives through the physical mapping.

#include <volume_data.hpp>

Public Attributes

- ElementBatch<::mfem::Vector > reference
- ElementBatch<::mfem::Vector > physical

10.346.1 Detailed Description

Reference samples remain available for derivatives through the physical mapping.

10.346.2 Member Data Documentation

10.346.2.1 physical

ElementBatch<::mfem::Vector> serif::operators::backend::mfem::OperandSamples::physical

10.346.2.2 reference

ElementBatch<::mfem::Vector> serif::operators::backend::mfem::OperandSamples::reference

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.347 serif::equations::OperandValues< OperandsT, OutputT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/equations/point.hpp

10.348 serif::operators::OperandValues< OperandsT, OutputT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/equations/point.hpp

10.349 serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT > Struct Template Reference

#include <point.hpp>

Public Types

- template<typename OperandT>
  using Entry = std::conditional_t<std::is_void_v<OutputT>, discretization::forms::OperandValue<OperandT>, discretization::forms::PartialDerivative<OutputT, discretization::forms::OperandValue<OperandT>>>
- using Storage = std::conditional_t<(std::same_as<Entry<Operands>, double> && ...), std::array<double, count>, std::tuple<Entry<Operands>...>>

Public Member Functions

- template<typename KeyT>
  decltype(auto) get ()

  Return the value or local partial for a uniquely identified operand.
- template<typename KeyT>
  decltype(auto) get () const

  Read the value or local partial without copying vector entries.

Static Public Member Functions

- template<typename KeyT>
  static consteval std::size_t index ()

  Lookup by full operand identity; a term alone is allowed only when unambiguous.

Public Attributes

- Storage values {}

Static Public Attributes

- static constexpr std::size_t count = sizeof...(Operands)

10.349.1 Member Typedef Documentation

10.349.1.1 Entry

template<typename OutputT, discretization::forms::IsOperand... Operands>
template<typename OperandT>
using serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >::Entry = std::conditional_t<std::is_void_v<OutputT>, discretization::forms::OperandValue<OperandT>, discretization::forms::PartialDerivative<OutputT, discretization::forms::OperandValue<OperandT>>>

10.349.1.2 Storage

template<typename OutputT, discretization::forms::IsOperand... Operands>
using serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >::Storage = std::conditional_t<(std::same_as<Entry<Operands>, double> && ...), std::array<double, count>, std::tuple<Entry<Operands>...>>

10.349.2 Member Function Documentation

10.349.2.1 get() [1/2]

template<typename OutputT, discretization::forms::IsOperand... Operands>
template<typename KeyT>
decltype(auto) serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >::get () [inline]

Return the value or local partial for a uniquely identified operand.

10.349.2.2 get() [2/2]

template<typename OutputT, discretization::forms::IsOperand... Operands>
template<typename KeyT>
decltype(auto) serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >::get () const [inline]

Read the value or local partial without copying vector entries.

10.349.2.3 index()

template<typename OutputT, discretization::forms::IsOperand... Operands>
template<typename KeyT>
static consteval std::size_t serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >::index () [inline], [static], [consteval]

Lookup by full operand identity; a term alone is allowed only when unambiguous.

10.349.3 Member Data Documentation

10.349.3.1 count

template<typename OutputT, discretization::forms::IsOperand... Operands>
std::size_t serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >::count = sizeof...(Operands) [static], [constexpr]

10.349.3.2 values

template<typename OutputT, discretization::forms::IsOperand... Operands>
Storage serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >::values {}

The documentation for this struct was generated from the following file:

- src/include/serif/equations/point.hpp

10.350 serif::discretization::forms::Operation Struct Reference

The Operation struct defines a set of operations that can be applied to terms in the context of a form. These operations include value, gradient, divergence, curl, and normal trace. Similar to the statement I made at the top of forms.hpp, if you are a developer looking to extend the mathematical capabilities of SERiF this is a place you should be looking early on.

#include <operands.hpp>

Classes

- struct Curl
  Operation tag for evaluating a curl.
- struct Divergence
  Operation tag for evaluating a divergence.
- struct Gradient
  Operation tag for evaluating a gradient.
- struct NormalTrace
  Operation tag for evaluating the normal flux along a boundary.
- struct Value
  Operation tag for evaluating a value.

10.350.1 Detailed Description

The Operation struct defines a set of operations that can be applied to terms in the context of a form. These operations include value, gradient, divergence, curl, and normal trace. Similar to the statement I made at the top of forms.hpp, if you are a developer looking to extend the mathematical capabilities of SERiF this is a place you should be looking early on.

Note

NormalTrace I find also to be somewhat confusing. Like many parts of the vocab at this level of SERiF this originates from finite element analysis (though I am sure they got it from somewhere else), this just means the normal fluxes along a boundary.

If you extend this struct you should also extend IsOperation (just below it in the file)

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/operands.hpp

10.351 serif::discretization::forms::OperationValue< Rank, OperationT > Struct Template Reference

General case for compile time metadata associating a tensor rank to an operation. The specialization of this will also specific what the return value of an operation of that rank will be.

10.351.1 Detailed Description

template<std::size_t Rank, typename OperationT>
struct serif::discretization::forms::OperationValue< Rank, OperationT >

General case for compile time metadata associating a tensor rank to an operation. The specialization of this will also specific what the return value of an operation of that rank will be.

Template Parameters

Rank	The rank of the input
OperationT	the operation to conduct on that input

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/value.hpp

10.352 serif::discretization::forms::OperationValue< 0, Operation::Gradient > Struct Reference

The gradient of a rank 0 tensor is a rank 1 tensor (a vector of doubles)

#include <value.hpp>

Public Types

- using type = VectorValue

10.352.1 Detailed Description

The gradient of a rank 0 tensor is a rank 1 tensor (a vector of doubles)

10.352.2 Member Typedef Documentation

10.352.2.1 type

using serif::discretization::forms::OperationValue< 0, Operation::Gradient >::type = VectorValue

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/value.hpp

10.353 serif::discretization::forms::OperationValue< 0, Operation::Value > Struct Reference

The value of a rank 0 tensor is a scalar, a double in the case of SERiF where we restrict ourselves to double precision.

#include <value.hpp>

Public Types

- using type = double

10.353.1 Detailed Description

The value of a rank 0 tensor is a scalar, a double in the case of SERiF where we restrict ourselves to double precision.

10.353.2 Member Typedef Documentation

10.353.2.1 type

using serif::discretization::forms::OperationValue< 0, Operation::Value >::type = double

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/value.hpp

10.354 serif::discretization::forms::OperationValue< 1, Operation::Curl > Struct Reference

The curl of a rank 1 tensor is a rank 1 tensor (a vector of doubles)

#include <value.hpp>

Public Types

- using type = VectorValue

10.354.1 Detailed Description

The curl of a rank 1 tensor is a rank 1 tensor (a vector of doubles)

10.354.2 Member Typedef Documentation

10.354.2.1 type

using serif::discretization::forms::OperationValue< 1, Operation::Curl >::type = VectorValue

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/value.hpp

10.355 serif::discretization::forms::OperationValue< 1, Operation::Divergence > Struct Reference

The divergence of a rank one tensor is a scalar (a double)

#include <value.hpp>

Public Types

- using type = double

10.355.1 Detailed Description

The divergence of a rank one tensor is a scalar (a double)

10.355.2 Member Typedef Documentation

10.355.2.1 type

using serif::discretization::forms::OperationValue< 1, Operation::Divergence >::type = double

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/value.hpp

10.356 serif::discretization::forms::OperationValue< 1, Operation::NormalTrace > Struct Reference

#include <value.hpp>

Public Types

- using type = double

10.356.1 Detailed Description

The normal trace (flux normal to a boundary surface) of a rank one tensor is a scalar (a double)

10.356.2 Member Typedef Documentation

10.356.2.1 type

using serif::discretization::forms::OperationValue< 1, Operation::NormalTrace >::type = double

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/value.hpp

10.357 serif::discretization::forms::OperationValue< 1, Operation::Value > Struct Reference

The value of a rank 1 tensor is a rank 1 tensor (a vector of doubles in SERiF)

#include <value.hpp>

Public Types

- using type = VectorValue

10.357.1 Detailed Description

The value of a rank 1 tensor is a rank 1 tensor (a vector of doubles in SERiF)

10.357.2 Member Typedef Documentation

10.357.2.1 type

using serif::discretization::forms::OperationValue< 1, Operation::Value >::type = VectorValue

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/value.hpp

10.358 serif::utils::dependencies::keys::Output Struct Reference

The main result of a computation.

#include <keys.hpp>

10.358.1 Detailed Description

The main result of a computation.

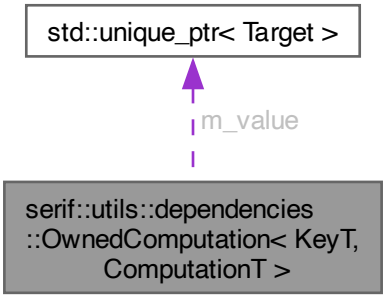
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/keys.hpp

10.359 serif::utils::dependencies::OwnedComputation< KeyT, ComputationT > Class Template Reference

#include <computation_store.hpp>

Collaboration diagram for serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >:



Public Types

- using Key = KeyT
- using Target = ComputationT

Public Member Functions

- template<typename ValueT>
  requires std::constructible_from<Target, ValueT>
  OwnedComputation (ValueT &&value)

  Give a cache or computation owner a stable address, independent of its store's location.
- OwnedComputation (const OwnedComputation &)=delete

  A computation has one owner; use references to share access to it.
- OwnedComputation & operator= (const OwnedComputation &)=delete
- OwnedComputation (OwnedComputation &&) noexcept=default

  Transfer ownership without moving the computation itself.
- OwnedComputation & operator= (OwnedComputation &&)=delete

  Replacing an entry could invalidate references held by prepared operators.
- Target & get () &

  Access the persistent computation, for example to prepare its next state.
- const Target & get () const &

  Read a computation through its owner without allowing it to change.
- Target & get () &&=delete

  References cannot safely borrow computations from temporary owners.
- const Target & get () const &&=delete

Private Member Functions

- Target & checked_value () const

  A moved-from entry no longer owns an object that a consumer can borrow.

Private Attributes

- std::unique_ptr< Target > m_value

10.359.1 Member Typedef Documentation

10.359.1.1 Key

template<typename KeyT, typename ComputationT>
using serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::Key = KeyT

10.359.1.2 Target

template<typename KeyT, typename ComputationT>
using serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::Target = ComputationT

10.359.2 Constructor & Destructor Documentation

10.359.2.1 OwnedComputation() [1/3]

template<typename KeyT, typename ComputationT>
template<typename ValueT>
requires std::constructible_from<Target, ValueT>
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::OwnedComputation (
ValueT && value) [inline], [explicit]

Give a cache or computation owner a stable address, independent of its store's location.

10.359.2.2 OwnedComputation() [2/3]

template<typename KeyT, typename ComputationT>
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::OwnedComputation (
const OwnedComputation< KeyT, ComputationT > &) [delete]

A computation has one owner; use references to share access to it.

10.359.2.3 OwnedComputation() [3/3]

template<typename KeyT, typename ComputationT>
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::OwnedComputation (
OwnedComputation< KeyT, ComputationT > &&) [default], [noexcept]

Transfer ownership without moving the computation itself.

10.359.3 Member Function Documentation

10.359.3.1 checked_value()

template<typename KeyT, typename ComputationT>
Target & serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::checked_value () const [inline], [nodiscard], [private]

A moved-from entry no longer owns an object that a consumer can borrow.

10.359.3.2 get() [1/4]

template<typename KeyT, typename ComputationT>
Target & serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::get () & [inline], [nodiscard]

Access the persistent computation, for example to prepare its next state.

10.359.3.3 get() [2/4]

template<typename KeyT, typename ComputationT>
Target & serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::get () && [delete]

References cannot safely borrow computations from temporary owners.

10.359.3.4 get() [3/4]

template<typename KeyT, typename ComputationT>
const Target & serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::get () const & [inline], [nodiscard]

Read a computation through its owner without allowing it to change.

10.359.3.5 get() [4/4]

template<typename KeyT, typename ComputationT>
const Target & serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::get () const && [delete]

10.359.3.6 operator=() [1/2]

template<typename KeyT, typename ComputationT>
OwnedComputation & serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::operator= (
const OwnedComputation< KeyT, ComputationT > &) [delete]

10.359.3.7 operator=() [2/2]

template<typename KeyT, typename ComputationT>
OwnedComputation & serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::operator= (
OwnedComputation< KeyT, ComputationT > &&) [delete]

Replacing an entry could invalidate references held by prepared operators.

10.359.4 Member Data Documentation

10.359.4.1 m_value

template<typename KeyT, typename ComputationT>
std::unique_ptr<Target> serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >::m_value [private]

The documentation for this class was generated from the following file:

- src/include/serif/utils/dependencies/computation_store.hpp

10.360 serif::utils::dependencies::detail::OwnedValue< T > Struct Template Reference

#include <resource.hpp>

Public Types

- using Target = T

Public Member Functions

- const T & get () const
  Expose a source for read-only computation.

Public Attributes

- T value

10.360.1 Member Typedef Documentation

10.360.1.1 Target

template<typename T>
using serif::utils::dependencies::detail::OwnedValue< T >::Target = T

10.360.2 Member Function Documentation

10.360.2.1 get()

template<typename T>
const T & serif::utils::dependencies::detail::OwnedValue< T >::get () const [inline], [nodiscard]

Expose a source for read-only computation.

10.360.3 Member Data Documentation

10.360.3.1 value

template<typename T>
T serif::utils::dependencies::detail::OwnedValue< T >::value

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/resource.hpp

10.361 serif::geometry::surface::IsobaricSurface::Parameters Struct Reference

#include <isobaric.hpp>

Public Attributes

- dimensions::PressureValue PSurf

10.361.1 Member Data Documentation

10.361.1.1 PSurf

dimensions::PressureValue serif::geometry::surface::IsobaricSurface::Parameters::PSurf

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/isobaric.hpp

10.362 serif::utils::dependencies::keys::Parameters Struct Reference

Configuration values that can affect the result.

#include <keys.hpp>

10.362.1 Detailed Description

Configuration values that can affect the result.

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/keys.hpp

10.363 serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity > Class Template Reference

Class to represent a compile time derivative relation between two quantities. Note that this class does not, in fact cannot, compute any derivatives itself. Rather, this is a way for the compiler to understand higher level mathematical relations. Unless you are really in the weeds of the development team you will likely not need to interact with this directly. Further note that this class does include (unlike the more basic dimension system) some basic dimensional analysis. Specifically, there are some overloaded operators that allow for certain specializations of PartialDerivative to operate on one and other conditioned on what their input and output types are. This does not do any thing like convert area to length^2. However, it will do more basic things like ensure that you can only add two partials which agree on their dimensions exactly. Or that while you can multiple a partial by any scalar, you cannot multiply arbitrary partials.

#include <partial.hpp>

Public Member Functions

- constexpr PartialDerivative (const double value) noexcept
- constexpr double value () const noexcept

  Get the underlying value as a compile time constant if possible. This allows the compiler, where the constructor may be determined at compile time, to fully optimize PartialDerivative operations as if they were doubles.

Private Attributes

- double m_value

  Numeric value associated with the partial. If this is known at compile time then the compiler can optimize operations with partial.value() to be equivalent to compile time double operations.

Friends

- constexpr PartialDerivative< OutputQuantity, InputQuantity > operator+ (const PartialDerivative< OutputQuantity, InputQuantity > &lhs, const PartialDerivative< OutputQuantity, InputQuantity > &rhs) noexcept

  Overload the + operator between two partials which share the exact same input and output types.
- constexpr PartialDerivative< OutputQuantity, InputQuantity > operator- (const PartialDerivative< OutputQuantity, InputQuantity > &lhs, const PartialDerivative< OutputQuantity, InputQuantity > &rhs) noexcept

  Overload the - operator between two partials which share the exact same input and output types.
- template<utils::misc::concepts::IsNumeric Scalar>
  constexpr PartialDerivative< OutputQuantity, InputQuantity > operator* (const PartialDerivative< OutputQuantity, InputQuantity > &lhs, const Scalar &rhs) noexcept

  Overload of the * operator for a left partial and a right scalar and output types.
- template<utils::misc::concepts::IsNumeric Scalar>
  constexpr PartialDerivative< OutputQuantity, InputQuantity > operator* (const Scalar &lhs, const PartialDerivative< OutputQuantity, InputQuantity > &rhs) noexcept

  Overload of the * operator for a right partial and a left scalar and output types.
- template<utils::misc::concepts::IsNumeric Scalar>
  constexpr PartialDerivative< OutputQuantity, InputQuantity > operator/ (const PartialDerivative< OutputQuantity, InputQuantity > &lhs, const Scalar &rhs) noexcept

  Overload of the / operator for a left partial and a right scalar and output types.
- template<utils::misc::concepts::IsNumeric Scalar>
  constexpr std::partial_ordering operator<=> (const PartialDerivative< OutputQuantity, Input↩ Quantity > &lhs, Scalar rhs) noexcept

  Partial ordering overload to compare a left partial and a right scalar.
- template<utils::misc::concepts::IsNumeric Scalar>
  constexpr std::partial_ordering operator<=> (Scalar lhs, const PartialDerivative< OutputQuantity, InputQuantity > &rhs) noexcept

  Partial ordering overload to compare a right partial and a left scalar.
- constexpr std::partial_ordering operator<=> (const PartialDerivative< OutputQuantity, Input↩ Quantity > &lhs, const PartialDerivative< OutputQuantity, InputQuantity > &rhs) noexcept

  Partial ordering comparing two partials with exactly the same input and output types.

10.363.1 Detailed Description

template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
class serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >

Class to represent a compile time derivative relation between two quantities. Note that this class does not, in fact cannot, compute any derivatives itself. Rather, this is a way for the compiler to understand higher level mathematical relations. Unless you are really in the weeds of the development team you will likely not need to interact with this directly. Further note that this class does include (unlike the more basic dimension system) some basic dimensional analysis. Specifically, there are some overloaded operators that allow for certain specializations of PartialDerivative to operate on one and other conditioned on what their input and output types are. This does not do any thing like convert area to length^2. However, it will do more basic things like ensure that you can only add two partials which agree on their dimensions exactly. Or that while you can multiple a partial by any scalar, you cannot multiply arbitrary partials.

Template Parameters

OutputQuantity	
InputQuantity	

10.363.2 Constructor & Destructor Documentation

10.363.2.1 PartialDerivative()

template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >::PartialDerivative (
const double value) [inline], [explicit], [constexpr], [noexcept]

10.363.3 Member Function Documentation

10.363.3.1 value()

template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
double serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >::value () const [inline], [nodiscard], [constexpr], [noexcept]

Get the underlying value as a compile time constant if possible. This allows the compiler, where the constructor may be determined at compile time, to fully optimize PartialDerivative operations as if they were doubles.

Note

calling value gets the underlying numeric value directly thereby bypassing the limitations on what a valid operation are (e.g. what the operator overloads enforce).

Returns

m_value: This is whatever was passed at constructor time since there is no way to modify this value after construction without some truly horrific C++ tricks

10.363.4 Friends And Related Symbol Documentation

10.363.4.1 operator∗ [1/2]

```
template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
template<utils::misc::concepts::IsNumeric Scalar>
PartialDerivative< OutputQuantity, InputQuantity > operator∗ (
            const PartialDerivative< OutputQuantity, InputQuantity > & lhs,
            const Scalar & rhs)   [friend]
```

Overload of the ∗ operator for a left partial and a right scalar and output types.

Template Parameters

Scalar	Any value satisfying the IsNumeric concept (integral types and float types)

Parameters

lhs	Partial with some input and some output type
rhs	Some numeric quantity

Returns

A new partial with the same input and output types where m_value = lhs.m_value ∗ rhs

10.363.4.2 operator∗ [2/2]

```
template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
template<utils::misc::concepts::IsNumeric Scalar>
PartialDerivative< OutputQuantity, InputQuantity > operator∗ (
            const Scalar & lhs,
            const PartialDerivative< OutputQuantity, InputQuantity > & rhs)   [friend]
```

Overload of the ∗ operator for a right partial and a left scalar and output types.

Template Parameters

Scalar	Any value satisfying the IsNumeric concept (integral types and float types)

Parameters

lhs	Some numeric quantity
rhs	Partial with some input and some output type

Returns

A new partial with the same input and output types where m_value = lhs ∗ rhs.m_value

10.363.4.3 operator+

```
template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
PartialDerivative< OutputQuantity, InputQuantity > operator+ (
                const PartialDerivative< OutputQuantity, InputQuantity > & lhs,
                const PartialDerivative< OutputQuantity, InputQuantity > & rhs)   [friend]
```

Overload the + operator between two partials which share the exact same input and output types.

Parameters

lhs	One partial
rhs	Other Partial

Returns

A new partial with the same input and output types where m_value = lhs.m_value + rhs.m_value

10.363.4.4 operator-

```
template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
PartialDerivative< OutputQuantity, InputQuantity > operator- (
                const PartialDerivative< OutputQuantity, InputQuantity > & lhs,
                const PartialDerivative< OutputQuantity, InputQuantity > & rhs)   [friend]
```

Overload the - operator between two partials which share the exact same input and output types.

Parameters

lhs	One partial
rhs	Other Partial

Returns

A new partial with the same input and output types where m_value = lhs.m_value - rhs.m_value

10.363.4.5 operator/

```
template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
template<utils::misc::concepts::IsNumeric Scalar>
PartialDerivative< OutputQuantity, InputQuantity > operator/ (
                const PartialDerivative< OutputQuantity, InputQuantity > & lhs,
                const Scalar & rhs)   [friend]
```

Overload of the / operator for a left partial and a right scalar and output types.

Template Parameters

Scalar	Any value satisfying the IsNumeric concept (integral types and float types)

Parameters

lhs	Partial with some input and some output type
rhs	Some numeric quantity

Returns

A new partial with the same input and output types where m_value = lhs.m_value / rhs

10.363.4.6 operator<=> [1/3]

```
template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
std::partial_ordering operator<=> (
        const PartialDerivative< OutputQuantity, InputQuantity > & lhs,
        const PartialDerivative< OutputQuantity, InputQuantity > & rhs)   [friend]
```

Partial ordering comparing two partials with exactly the same input and output types.

Parameters

lhs	Some Partial
rhs	Other Partial

Returns

Partial ordering comparing lhs.m_value to rhs.m_value

10.363.4.7 operator<=> [2/3]

```
template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
template<utils::misc::concepts::IsNumeric Scalar>
std::partial_ordering operator<=> (
        const PartialDerivative< OutputQuantity, InputQuantity > & lhs,
        Scalar rhs)   [friend]
```

Partial ordering overload to compare a left partial and a right scalar.

Template Parameters

Scalar	Any value satisfying the IsNumeric concept (integral types and float types)

Parameters

lhs	Some numeric quantity
rhs	Some partial with an input and output type

Returns

partial ordering comparing rhs.m_value to lhs

10.363.4.8 operator<=> [3/3]

```
template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
template<utils::misc::concepts::IsNumeric Scalar>
std::partial_ordering operator<=> (
            Scalar lhs,
            const PartialDerivative< OutputQuantity, InputQuantity > & rhs)    [friend]
```

Partial ordering overload to compare a right partial and a left scalar.

Template Parameters

Scalar	Any value satisfying the IsNumeric concept (integral types and float types)

Parameters

lhs	Some numeric quantity
rhs	Some partial with an input and output type

Returns

partial ordering comparing rhs.m_value to lhs

10.363.5 Member Data Documentation

10.363.5.1 m_value

```
template<ThermodynamicQuantityType OutputQuantity, ThermodynamicQuantityType InputQuantity>
double serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >::m_value    [private]
```

Numeric value associated with the partial. If this is known at compile time then the compiler can optimize operations with partial.value() to be equivalent to compile time double operations.

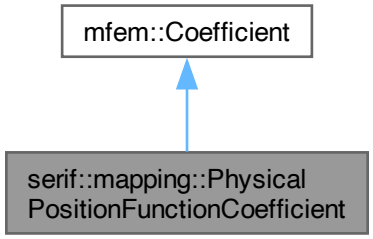
The documentation for this class was generated from the following file:

- src/include/serif/dimensions/partial.hpp

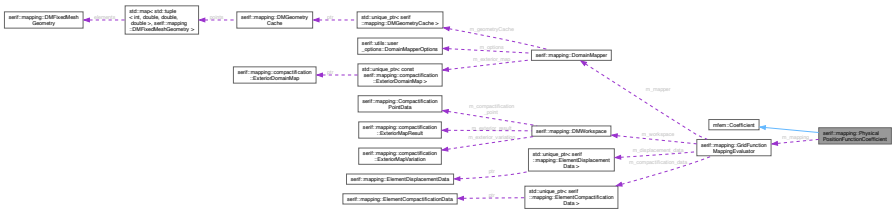
10.364 serif::mapping::PhysicalPositionFunctionCoefficient Class Reference

#include <coefficients.hpp>

Inheritance diagram for serif::mapping::PhysicalPositionFunctionCoefficient:



Collaboration diagram for serif::mapping::PhysicalPositionFunctionCoefficient:



Public Types

- using Func = std::function<double(const mfem::Vector& x)>

Public Member Functions

- PhysicalPositionFunctionCoefficient (const DomainMapper &mapper, const mfem::GridFunction &displacement, const mfem::GridFunction &compactification_coordinate, Func func)
- double Eval (mfem::ElementTransformation &transformation, const mfem::IntegrationPoint &ip) override

Private Attributes

- Func m_f
- GridFunctionMappingEvaluator m_mapping

10.364.1 Member Typedef Documentation

10.364.1.1 Func

using serif::mapping::PhysicalPositionFunctionCoefficient::Func = std::function<double(const mfem::Vector& x)>

10.364.2 Constructor & Destructor Documentation

10.364.2.1 PhysicalPositionFunctionCoefficient()

serif::mapping::PhysicalPositionFunctionCoefficient::PhysicalPositionFunctionCoefficient (
const DomainMapper & mapper,
const mfem::GridFunction & displacement,
const mfem::GridFunction & compactification_coordinate,
Func func)

10.364.3 Member Function Documentation

10.364.3.1 Eval()

double serif::mapping::PhysicalPositionFunctionCoefficient::Eval (
mfem::ElementTransformation & transformation,
const mfem::IntegrationPoint & ip) [override]

10.364.4 Member Data Documentation

10.364.4.1 m_f

Func serif::mapping::PhysicalPositionFunctionCoefficient::m_f [private]

10.364.4.2 m_mapping

GridFunctionMappingEvaluator serif::mapping::PhysicalPositionFunctionCoefficient::m_mapping [private]

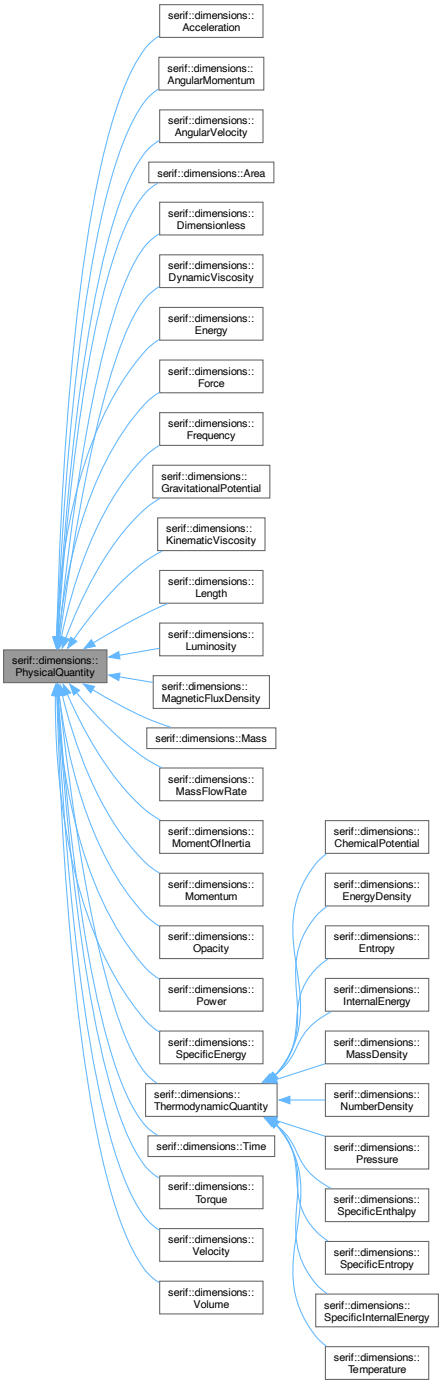
The documentation for this class was generated from the following files:

- src/include/serif/mapping/coefficients.hpp
- src/lib/serif/mapping/coefficients.cpp

10.365 serif::dimensions::PhysicalQuantity Struct Reference

#include <quantities.hpp>

Inheritance diagram for serif::dimensions::PhysicalQuantity:



10.365.1 Detailed Description

@breif General type for any physical quantity.

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/quantities.hpp

10.366 serif::operators::backend::mfem::volume_keys::PhysicalValues< Contribution, OperandT > Struct Template Reference

#include <volume_data.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.367 serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT > Struct Template Reference

Store the scalar and point linearization together so a result cannot mix accepted states.

#include <nodal_data.hpp>

Public Types

- using Border = typename EquationT::Border

Public Attributes

- NodalData< EquationT, SchemaT > nodal
- const PointBorderDiscretization< EquationT, SchemaT >::BorderBinding & border_binding
- double border_coefficient

10.367.1 Detailed Description

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
struct serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >

Store the scalar and point linearization together so a result cannot mix accepted states.

10.367.2 Member Typedef Documentation

10.367.2.1 Border

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
using serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >::Border = typename EquationT::Border

10.367.3 Member Data Documentation

10.367.3.1 border_binding

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
const PointBorderDiscretization<EquationT,SchemaT>::BorderBinding& serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >::border_binding

10.367.3.2 border_coefficient

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
double serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >::border_coefficient

10.367.3.3 nodal

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
NodalData<EquationT, SchemaT> serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >::nodal

The documentation for this struct was generated from the following file:

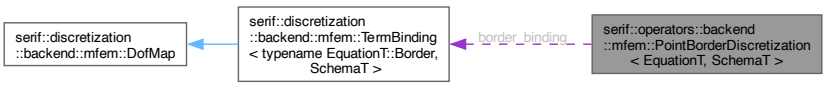
- src/include/serif/operators/backend/mfem/nodal_data.hpp

10.368 serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT > Struct Template Reference

A point condition and its scalar border must use the same mesh.

#include <nodal_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >:



Public Types

- using Equation = EquationT
- using BorderBinding = discretization::backend::mfem::TermBinding<typename EquationT::Border, SchemaT>

Public Attributes

- NodalDiscretization< EquationT, SchemaT > nodal
- const BorderBinding & border_binding

10.368.1 Detailed Description

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
struct serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >

A point condition and its scalar border must use the same mesh.

10.368.2 Member Typedef Documentation

10.368.2.1 BorderBinding

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
using serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >::BorderBinding = discretization::backend::mfem::TermBi
EquationT::Border, SchemaT>

10.368.2.2 Equation

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
using serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >::Equation = EquationT

10.368.3 Member Data Documentation

10.368.3.1 border_binding

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
const BorderBinding& serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >::border_binding

10.368.3.2 nodal

template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
NodalDiscretization<EquationT, SchemaT> serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT
>::nodal

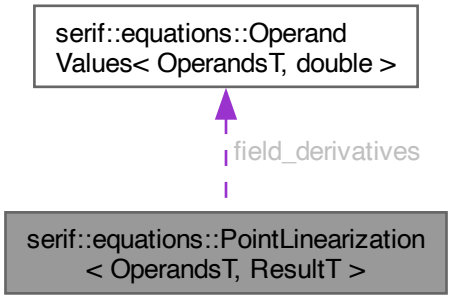
The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/nodal_data.hpp

10.369 serif::equations::PointLinearization< OperandsT, ResultT > Struct Template Reference

#include <point.hpp>

Collaboration diagram for serif::equations::PointLinearization< OperandsT, ResultT >:



Public Attributes

- ResultT residual {}
- OperandValues< OperandsT, ResultT > field_derivatives
- discretization::forms::PartialDerivative< ResultT, discretization::forms::VectorValue > position_derivative {}

10.369.1 Member Data Documentation

10.369.1.1 field_derivatives

template<typename OperandsT, typename ResultT = double>
OperandValues<OperandsT, ResultT> serif::equations::PointLinearization< OperandsT, ResultT >::field_derivatives

10.369.1.2 position_derivative

template<typename OperandsT, typename ResultT = double>
discretization::forms::PartialDerivative<ResultT, discretization::forms::VectorValue> serif::equations::PointLinearization< OperandsT, ResultT >::position_derivative {}

10.369.1.3 residual

template<typename OperandsT, typename ResultT = double>
ResultT serif::equations::PointLinearization< OperandsT, ResultT >::residual {}

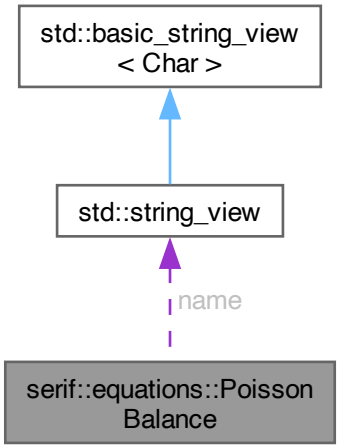
The documentation for this struct was generated from the following file:

- src/include/serif/equations/point.hpp

10.370 serif::equations::PoissonBalance Class Reference

#include <mixed_gravity.hpp>

Collaboration diagram for serif::equations::PoissonBalance:



Public Types

- using Test = discretization::blocks::gravity::potential

Public Member Functions

- PoissonBalance (double four_pi_G)
- auto contributions () const

Static Public Attributes

- static constexpr std::string_view name = "Poisson balance"

Private Attributes

- double m_four_pi_G

10.370.1 Member Typedef Documentation

10.370.1.1 Test

using serif::equations::PoissonBalance::Test = discretization::blocks::gravity::potential

10.370.2 Constructor & Destructor Documentation

10.370.2.1 PoissonBalance()

serif::equations::PoissonBalance::PoissonBalance (
double four_pi_G) [inline], [explicit]

10.370.3 Member Function Documentation

10.370.3.1 contributions()

auto serif::equations::PoissonBalance::contributions () const [inline]

10.370.4 Member Data Documentation

10.370.4.1 m_four_pi_G

double serif::equations::PoissonBalance::m_four_pi_G [private]

10.370.4.2 name

std::string_view serif::equations::PoissonBalance::name = "Poisson balance" [static], [constexpr]

The documentation for this class was generated from the following file:

- src/include/serif/equations/mixed_gravity.hpp

10.371 serif::discretization::quadrature::Policy Class Reference

#include <policy.hpp>

Collaboration diagram for serif::discretization::quadrature::Policy:



Public Member Functions

- Policy (RuleSet ruleSet)
- Resolution resolve (const Query &query) const

Private Member Functions

- const RuleControl & get_role_control (QuadratureRole role) const

Static Private Member Functions

- static int compute_base_order (const Query &query)

Private Attributes

- RuleSet m_ruleSet

10.371.1 Constructor & Destructor Documentation

10.371.1.1 Policy()

serif::discretization::quadrature::Policy::Policy (
RuleSet ruleSet) [explicit]

10.371.2 Member Function Documentation

10.371.2.1 compute_base_order()

int serif::discretization::quadrature::Policy::compute_base_order (
const Query & query) [static], [nodiscard], [private]

10.371.2.2 get_role_control()

const RuleControl & serif::discretization::quadrature::Policy::get_role_control (
QuadratureRole role) const [nodiscard], [private]

10.371.2.3 resolve()

Resolution serif::discretization::quadrature::Policy::resolve (
const Query & query) const [nodiscard]

10.371.3 Member Data Documentation

10.371.3.1 m_ruleSet

RuleSet serif::discretization::quadrature::Policy::m_ruleSet [private]

The documentation for this class was generated from the following files:

- src/include/serif/discretization/quadrature/policy.hpp
- src/lib/serif/discretization/quadrature/policy.cpp

10.372 serif::eos::models::Polytrope Class Reference

#include <polytropic.hpp>

Public Types

- using Relations

Public Member Functions

- Polytrope (const PolytropeParameters &parameters)
- Polytrope (double n, double K)
- double polytropic_index () const noexcept
- double polytropic_constant () const noexcept
- double enthalpy_scale () const noexcept
- dimensions::PressureValue evaluate (relations::PressureFromDensity, dimensions::DensityValue density) const
- dimensions::PressureValue evaluate (relations::PressureFromSpecificEnthalpy, dimensions::SpecificEnthalpyValue specific_enthalpy) const
- dimensions::SpecificEnthalpyValue evaluate (relations::SpecificEnthalpyFromDensity, dimensions::DensityValue density) const
- dimensions::SpecificEnthalpyValue evaluate (relations::SpecificEnthalpyFromPressure, dimensions::PressureValue pressure) const

- dimensions::DensityValue evaluate (relations::DensityFromSpecificEnthalpy, dimensions::SpecificEnthalpyValue specific_enthalpy) const
- dimensions::PartialDerivative< dimensions::MassDensity, dimensions::SpecificEnthalpy > partial_derivative (relations::DensityFromSpecificEnthalpy, dimensions::WithRespectTo< dimensions::SpecificEnthalpy >, dimensions::SpecificEnthalpyValue specific_enthalpy) const
- dimensions::PartialDerivative< dimensions::Pressure, dimensions::SpecificEnthalpy > partial_derivative (relations::PressureFromSpecificEnthalpy, dimensions::WithRespectTo< dimensions::SpecificEnthalpy >, dimensions::SpecificEnthalpyValue specific_enthalpy) const
- dimensions::PartialDerivative< dimensions::Pressure, dimensions::MassDensity > partial_derivative (relations::PressureFromDensity, dimensions::WithRespectTo< dimensions::MassDensity >, dimensions::DensityValue density) const

Private Attributes

- double m_polytropic_index
- double m_polytropic_constant
- double m_enthalpy_scale

10.372.1 Member Typedef Documentation

10.372.1.1 Relations

using serif::eos::models::Polytrope::Relations

Initial value:

```
relations::RelationCatalog<
        relations::PressureFromDensity,
        relations::PressureFromSpecificEnthalpy,
        relations::SpecificEnthalpyFromDensity,
        relations::SpecificEnthalpyFromPressure,
        relations::DensityFromSpecificEnthalpy
    >
```

10.372.2 Constructor & Destructor Documentation

10.372.2.1 Polytrope() [1/2]

serif::eos::models::Polytrope::Polytrope (
const PolytropeParameters & parameters) [explicit]

10.372.2.2 Polytrope() [2/2]

serif::eos::models::Polytrope::Polytrope (
double n,
double K) [explicit]

10.372.3 Member Function Documentation

10.372.3.1 enthalpy_scale()

double serif::eos::models::Polytrope::enthalpy_scale () const [nodiscard], [noexcept]

10.372.3.2 evaluate() [1/5]

DensityValue serif::eos::models::Polytrope::evaluate (
relations::DensityFromSpecificEnthalpy ,
dimensions::SpecificEnthalpyValue specific_enthalpy) const [nodiscard]

10.372.3.3 evaluate() [2/5]

PressureValue serif::eos::models::Polytrope::evaluate (
relations::PressureFromDensity ,
dimensions::DensityValue density) const [nodiscard]

10.372.3.4 evaluate() [3/5]

PressureValue serif::eos::models::Polytrope::evaluate (
relations::PressureFromSpecificEnthalpy ,
dimensions::SpecificEnthalpyValue specific_enthalpy) const [nodiscard]

10.372.3.5 evaluate() [4/5]

SpecificEnthalpyValue serif::eos::models::Polytrope::evaluate (
relations::SpecificEnthalpyFromDensity ,
dimensions::DensityValue density) const [nodiscard]

10.372.3.6 evaluate() [5/5]

SpecificEnthalpyValue serif::eos::models::Polytrope::evaluate (
relations::SpecificEnthalpyFromPressure ,
dimensions::PressureValue pressure) const [nodiscard]

10.372.3.7 partial_derivative() [1/3]

PartialDerivative< MassDensity, SpecificEnthalpy > serif::eos::models::Polytrope::partial_derivative (
relations::DensityFromSpecificEnthalpy ,
dimensions::WithRespectTo< dimensions::SpecificEnthalpy > ,
dimensions::SpecificEnthalpyValue specific_enthalpy) const [nodiscard]

10.372.3.8 partial_derivative() [2/3]

PartialDerivative< Pressure, MassDensity > serif::eos::models::Polytrope::partial_derivative (
relations::PressureFromDensity ,
dimensions::WithRespectTo< dimensions::MassDensity > ,
dimensions::DensityValue density) const [nodiscard]

10.372.3.9 partial_derivative() [3/3]

PartialDerivative< Pressure, SpecificEnthalpy > serif::eos::models::Polytrope::partial_derivative (
relations::PressureFromSpecificEnthalpy ,
dimensions::WithRespectTo< dimensions::SpecificEnthalpy > ,
dimensions::SpecificEnthalpyValue specific_enthalpy) const [nodiscard]

10.372.3.10 polytropic_constant()

double serif::eos::models::Polytrope::polytropic_constant () const [nodiscard], [noexcept]

10.372.3.11 polytropic_index()

double serif::eos::models::Polytrope::polytropic_index () const [nodiscard], [noexcept]

10.372.4 Member Data Documentation

10.372.4.1 m_enthalpy_scale

double serif::eos::models::Polytrope::m_enthalpy_scale [private]

10.372.4.2 m_polytropic_constant

double serif::eos::models::Polytrope::m_polytropic_constant [private]

10.372.4.3 m_polytropic_index

double serif::eos::models::Polytrope::m_polytropic_index [private]

The documentation for this class was generated from the following files:

- src/include/serif/eos/models/polytropic.hpp
- src/lib/serif/eos/models/polytropic.cpp

10.373 serif::eos::models::PolytropeParameters Struct Reference

#include <polytropic.hpp>

Public Attributes

- double n
- double K

10.373.1 Member Data Documentation

10.373.1.1 K

double serif::eos::models::PolytropeParameters::K

10.373.1.2 n

double serif::eos::models::PolytropeParameters::n

The documentation for this struct was generated from the following file:

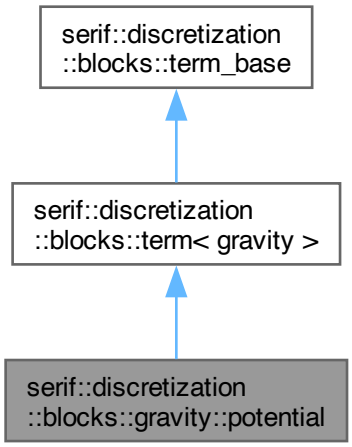
- src/include/serif/eos/models/polytropic.hpp

10.374 serif::discretization::blocks::gravity::potential Struct Reference

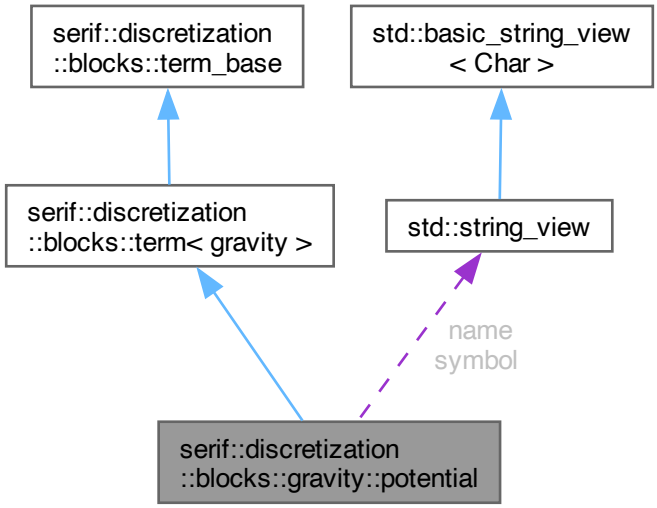
The gravitational potential term is a 2nd order L2 scalar field that is used to represent the gravitational potential of the system. Note that this is not exactly the same as the if you were to take the scalar potential of the gravitational gradient term. These are solved for by the code separately and weakly enforced to agree. Generally if you need the most accurate gradient you should use the gradient term and if you need the most accurate potential you should use this term (this is because differentiating this term loses an order of accuracy in the differentiation).

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::gravity::potential:



Collaboration diagram for serif::discretization::blocks::gravity::potential:



Classes

- struct residual

The residual block is a block that contains the residual of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system.

- struct value

  The value block is a block that contains the value of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system.

Public Types

- using FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::L2, 2, 0>

  2nd order L2 scalar field. Read the template parameters as follows: 1) L2 space, 2) 2nd order, 3) rank 0 tensor field
- using Support = domain::AllDomains

  We support gravity on all domains, primarily since the only real boundary condition we have in the general case is that the potential goes to zero at infinity, which is in the vacuum domain.
- using Quantity = dimensions::GravitationalPotential

  This is a dimension of gravitational potential, which is a scalar field that is associated with the gravitational potential of the system.

Public Types inherited from serif::discretization::blocks::term< gravity >

- using field_type

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<potential>()

  Compile time reflected name of the term.
- static constexpr std::string_view symbol = " "

  Compile time symbol used for the term. Note that this is a UTF-8 character and may not render correctly in all environments. (TODO: Should we introduce an ASCII fallback?, something like make this a tuple of form <main, fallback>?)

10.374.1 Detailed Description

The gravitational potential term is a 2nd order L2 scalar field that is used to represent the gravitational potential of the system. Note that this is not exactly the same as the if you were to take the scalar potential of the gravitational gradient term. These are solved for by the code separately and weakly enforced to agree. Generally if you need the most accurate gradient you should use the gradient term and if you need the most accurate potential you should use this term (this is because differentiating this term loses an order of accuracy in the differentiation).

10.374.2 Member Typedef Documentation

10.374.2.1 FunctionalSpace

using serif::discretization::blocks::gravity::potential::FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::L2, 2, 0>

2nd order L2 scalar field. Read the template parameters as follows: 1) L2 space, 2) 2nd order, 3) rank 0 tensor field

10.374.2.2 Quantity

using serif::discretization::blocks::gravity::potential::Quantity = dimensions::GravitationalPotential

This is a dimension of gravitational potential, which is a scalar field that is associated with the gravitational potential of the system.

10.374.2.3 Support

using serif::discretization::blocks::gravity::potential::Support = domain::AllDomains

We support gravity on all domains, primarily since the only real boundary condition we have in the general case is that the potential goes to zero at infinity, which is in the vacuum domain.

10.374.3 Member Data Documentation

10.374.3.1 name

std::string_view serif::discretization::blocks::gravity::potential::name = meta::type_name<potential>() [static], [constexpr]

Compile time reflected name of the term.

10.374.3.2 symbol

std::string_view serif::discretization::blocks::gravity::potential::symbol = " " [static], [constexpr]

Compile time symbol used for the term. Note that this is a UTF-8 character and may not render correctly in all environments. (TODO: Should we introduce an ASCII fallback?, something like make this a tuple of form <main, fallback>?)

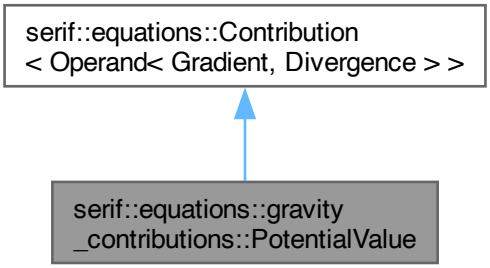
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

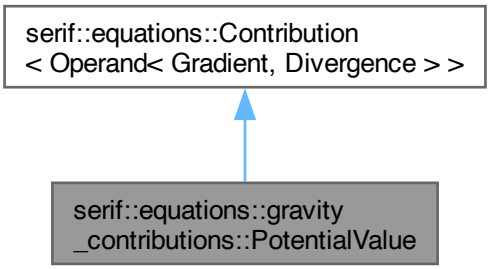
10.375 serif::equations::gravity_contributions::PotentialValue Struct Reference

#include <mixed_gravity.hpp>

Inheritance diagram for serif::equations::gravity_contributions::PotentialValue:



Collaboration diagram for serif::equations::gravity_contributions::PotentialValue:



Public Types

- using Operands = utils::types::TypeList<Operand<Potential>>

Public Types inherited from serif::equations::Contribution< Operand< Gradient, Divergence > >

- using Test
- using TestOperand
- using Support

Public Member Functions

- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const
  Integration by parts gives + div(v) Phi in the gravity-gradient row.
- int additional_order (Operand< Potential >, int) const
  The contribution is linear in potential.

10.375.1 Member Typedef Documentation

10.375.1.1 Operands

using serif::equations::gravity_contributions::PotentialValue::Operands = utils::types::TypeList<Operand<Potential>>

10.375.2 Member Function Documentation

10.375.2.1 additional_order()

int serif::equations::gravity_contributions::PotentialValue::additional_order (
Operand< Potential > ,
int) const [inline]

The contribution is linear in potential.

10.375.2.2 evaluate()

PointLinearization< Operands > serif::equations::gravity_contributions::PotentialValue::evaluate (
const EquationPoint< Operands > & point) const [inline]

Integration by parts gives + div(v) Phi in the gravity-gradient row.

The documentation for this struct was generated from the following file:

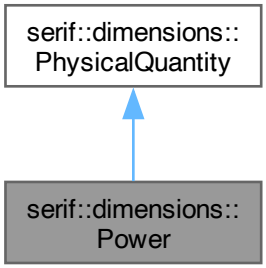
- src/include/serif/equations/mixed_gravity.hpp

10.376 serif::dimensions::Power Struct Reference

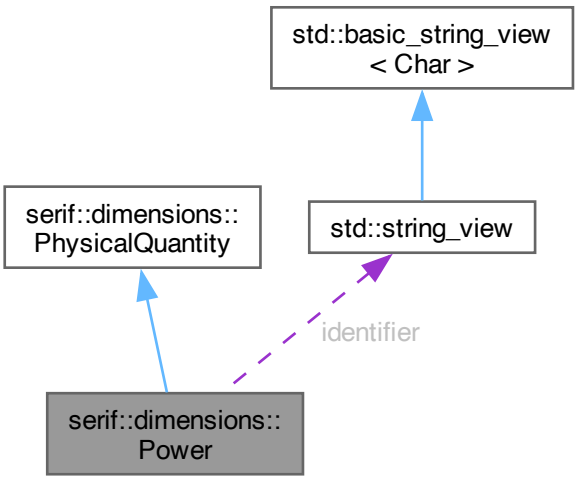
Dimension of Power.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Power:



Collaboration diagram for serif::dimensions::Power:



Static Public Attributes

- static constexpr std::string_view identifier = "power"

10.376.1 Detailed Description

Dimension of Power.

10.376.2 Member Data Documentation

10.376.2.1 identifier

std::string_view serif::dimensions::Power::identifier = "power" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.377 serif::geometry::deformation::PowerLawRadialExtension Struct Reference

Extend surface motion toward a fixed stellar center using a power of logical reference radius.

#include <models.hpp>

Public Member Functions

- bool operator== (const PowerLawRadialExtension &) const =default
  The radial power changes the interior response even for unchanged surface motion.

Public Attributes

- double radial_power {2.0}

10.377.1 Detailed Description

Extend surface motion toward a fixed stellar center using a power of logical reference radius.

10.377.2 Member Function Documentation

10.377.2.1 operator==()

bool serif::geometry::deformation::PowerLawRadialExtension::operator== (
const PowerLawRadialExtension &) const [nodiscard], [default]

The radial power changes the interior response even for unchanged surface motion.

10.377.3 Member Data Documentation

10.377.3.1 radial_power

double serif::geometry::deformation::PowerLawRadialExtension::radial_power {2.0}

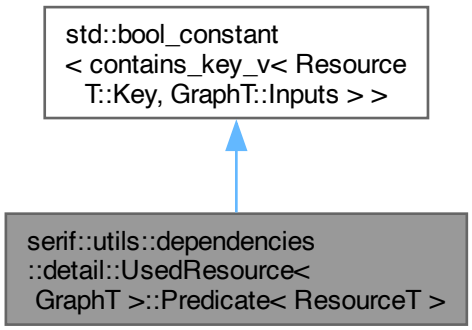
The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/models.hpp

10.378 serif::utils::dependencies::detail::UsedResource< GraphT >::Predicate< ResourceT > Struct Template Reference

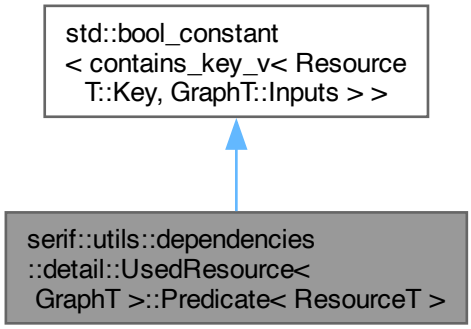
#include <cache_traits.hpp>

Inheritance diagram for serif::utils::dependencies::detail::UsedResource< GraphT >::Predicate< ResourceT >:



Collaboration diagram for serif::utils::dependencies::detail::UsedResource< GraphT >::Predicate< Re-

sourceT >:



The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_traits.hpp

10.379 serif::utils::dependencies::PreparationFailure< ErrorT > Struct Template Reference

#include <preparation.hpp>

Public Attributes

- std::size_t computation_index
- ErrorT error

10.379.1 Member Data Documentation

10.379.1.1 computation_index

template<typename ErrorT>
std::size_t serif::utils::dependencies::PreparationFailure< ErrorT >::computation_index

10.379.1.2 error

template<typename ErrorT>
ErrorT serif::utils::dependencies::PreparationFailure< ErrorT >::error

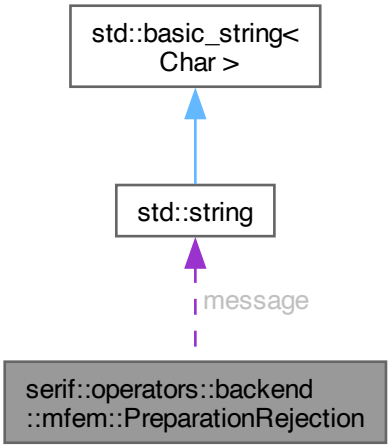
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/preparation.hpp

10.380 serif::operators::backend::mfem::PreparationRejection Struct Reference

#include <rejection.hpp>

Collaboration diagram for serif::operators::backend::mfem::PreparationRejection:



Public Attributes

- PreparationRejectionReason reason
- mapping::MappingStatus mapping_status {mapping::MappingStatus::VALID}
- int element {-1}
- int quadrature_point {-1}
- std::string message

10.380.1 Member Data Documentation

10.380.1.1 element

int serif::operators::backend::mfem::PreparationRejection::element {-1}

10.380.1.2 mapping_status

mapping::MappingStatus serif::operators::backend::mfem::PreparationRejection::mapping_status {mapping::MappingStatus::VALID}

10.380.1.3 message

std::string serif::operators::backend::mfem::PreparationRejection::message

10.380.1.4 quadrature_point

int serif::operators::backend::mfem::PreparationRejection::quadrature_point {-1}

10.380.1.5 reason

PreparationRejectionReason serif::operators::backend::mfem::PreparationRejection::reason

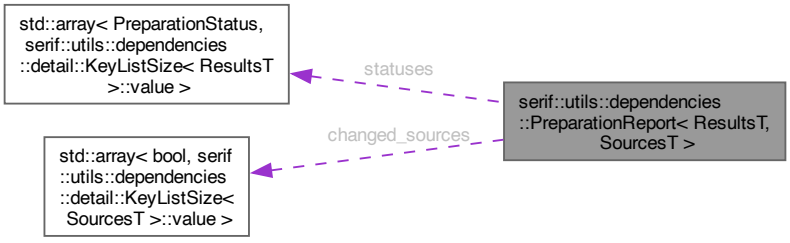
The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/rejection.hpp

10.381 serif::utils::dependencies::PreparationReport< ResultsT, SourcesT > Struct Template Reference

#include <preparation.hpp>

Collaboration diagram for serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >:



Public Member Functions

- template<typename KeyT>
  requires detail::contains_key_v<KeyT, ResultsT>
  PreparationStatus status () const

  Inspect a computation by its result key rather than an implementation index.
- template<typename KeyT>
  requires detail::contains_key_v<KeyT, SourcesT>
  bool source_changed () const

  An unused source is reported unchanged because it was not inspected by this request.
- std::size_t computed_count () const

  Count only computations evaluated during this successful preparation.
- std::size_t reused_count () const

  Count requested computations served from the accepted cache.

Public Attributes

- std::array< PreparationStatus, detail::KeyListSize< ResultsT >::value > statuses {}
- std::array< bool, detail::KeyListSize< SourcesT >::value > changed_sources {}

10.381.1 Member Function Documentation

10.381.1.1 computed_count()

template<IsKeyList ResultsT, IsKeyList SourcesT>
std::size_t serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >::computed_count () const [inline], [nodiscard]

Count only computations evaluated during this successful preparation.

10.381.1.2 reused_count()

template<IsKeyList ResultsT, IsKeyList SourcesT>
std::size_t serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >::reused_count () const [inline], [nodiscard]

Count requested computations served from the accepted cache.

10.381.1.3 source_changed()

template<IsKeyList ResultsT, IsKeyList SourcesT>
template<typename KeyT>
requires detail::contains_key_v<KeyT, SourcesT>
bool serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >::source_changed () const [inline], [nodiscard]

An unused source is reported unchanged because it was not inspected by this request.

10.381.1.4 status()

template<IsKeyList ResultsT, IsKeyList SourcesT>
template<typename KeyT>
requires detail::contains_key_v<KeyT, ResultsT>
PreparationStatus serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >::status () const [inline], [nodiscard]

Inspect a computation by its result key rather than an implementation index.

10.381.2 Member Data Documentation

10.381.2.1 changed_sources

template<IsKeyList ResultsT, IsKeyList SourcesT>
std::array<bool, detail::KeyListSize<SourcesT>::value> serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >↩
::changed_sources {}

10.381.2.2 statuses

template<IsKeyList ResultsT, IsKeyList SourcesT>
std::array<PreparationStatus, detail::KeyListSize<ResultsT>::value> serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >::statuses {}

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/preparation.hpp

10.382 serif::utils::misc::concepts::Prepend< T, List > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/utils/misc/concepts/filter.hpp

10.383 serif::utils::misc::concepts::Prepend< T, types::TypeList< Ts... > > Struct Template Reference

#include <filter.hpp>

Public Types

- using Type = types::TypeList<T, Ts...>

10.383.1 Member Typedef Documentation

10.383.1.1 Type

template<typename T, typename... Ts>
using serif::utils::misc::concepts::Prepend< T, types::TypeList< Ts... > >::Type = types::TypeList<T, Ts...>

The documentation for this struct was generated from the following file:

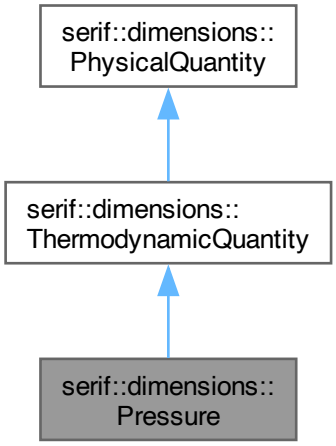
- src/include/serif/utils/misc/concepts/filter.hpp

10.384 serif::dimensions::Pressure Struct Reference

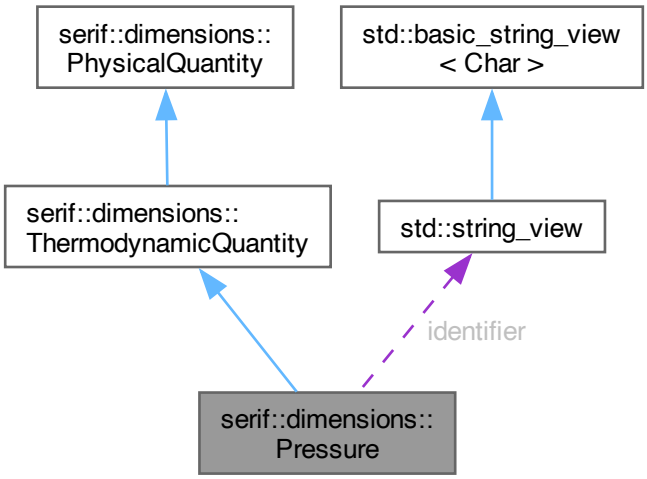
Dimension of Pressure.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Pressure:



Collaboration diagram for serif::dimensions::Pressure:



Static Public Attributes

- static constexpr std::string_view identifier = "pressure"

10.384.1 Detailed Description

Dimension of Pressure.

10.384.2 Member Data Documentation

10.384.2.1 identifier

std::string_view serif::dimensions::Pressure::identifier = "pressure" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.385 serif::geometry::surface::keys::PressureCondition Struct Reference

#include <samples.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/samples.hpp

10.386 serif::geometry::surface::detail::PressureConditionTracking Struct Reference

#include <samples.hpp>

Public Member Functions

- dimensions::PressureValue capture (const IsobaricSurface &condition) const
  Only prescribed pressure can affect the isobaric calculation.
- bool matches (const dimensions::PressureValue captured, const IsobaricSurface &condition) const
  Compare pressure exactly, retaining the dependency engine's normal change semantics.
- const void * identity (const IsobaricSurface &) const
  An equivalent pressure condition may be supplied by a newly constructed context.

10.386.1 Member Function Documentation

10.386.1.1 capture()

dimensions::PressureValue serif::geometry::surface::detail::PressureConditionTracking::capture (
const IsobaricSurface & condition) const [inline], [nodiscard]

Only prescribed pressure can affect the isobaric calculation.

10.386.1.2 identity()

const void * serif::geometry::surface::detail::PressureConditionTracking::identity (
const IsobaricSurface &) const [inline], [nodiscard]

An equivalent pressure condition may be supplied by a newly constructed context.

10.386.1.3 matches()

bool serif::geometry::surface::detail::PressureConditionTracking::matches (
const dimensions::PressureValue captured,
const IsobaricSurface & condition) const [inline], [nodiscard]

Compare pressure exactly, retaining the dependency engine's normal change semantics.

The documentation for this struct was generated from the following file:

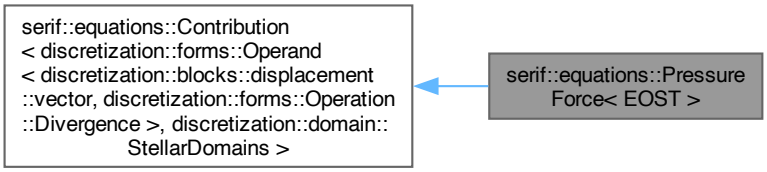
- src/include/serif/geometry/surface/samples.hpp

10.387 serif::equations::PressureForce< EOST > Class Template Reference

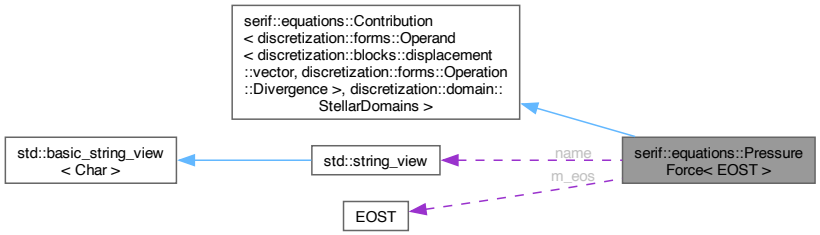
Pressure's contribution to displacement balance: -integral_star P(h) div(w) dV.

#include <pressure_force.hpp>

Inheritance diagram for serif::equations::PressureForce< EOST >:



Collaboration diagram for serif::equations::PressureForce< EOST >:



Public Types

- using Enthalpy = discretization::blocks::enthalpy::specific
- using EnthalpyValue = discretization::forms::Operand<Enthalpy>
- using Operands = utils::types::TypeList<EnthalpyValue>

Public Types inherited from serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::displacement::vecto

- using Test
- using TestOperand
- using Support

Public Member Functions

- PressureForce (EOST equation_of_state, const double enthalpy_power)
  The effective power describes P(h), not rho(h), for quadrature selection.
- PressureForce (EOST equation_of_state)
- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const
- int additional_order (EnthalpyValue, int) const

Static Public Attributes

- static constexpr std::string_view name = "pressure force"

Private Attributes

- EOST m_eos
- double m_enthalpy_power

10.387.1 Detailed Description

template<eos::models::PressureForceEOS EOST>
class serif::equations::PressureForce< EOST >

Pressure's contribution to displacement balance: -integral_star P(h) div(w) dV.

10.387.2 Member Typedef Documentation

10.387.2.1 Enthalpy

template<eos::models::PressureForceEOS EOST>
using serif::equations::PressureForce< EOST >::Enthalpy = discretization::blocks::enthalpy::specific

10.387.2.2 EnthalpyValue

template<eos::models::PressureForceEOS EOST>
using serif::equations::PressureForce< EOST >::EnthalpyValue = discretization::forms::Operand<Enthalpy>

10.387.2.3 Operands

template<eos::models::PressureForceEOS EOST>
using serif::equations::PressureForce< EOST >::Operands = utils::types::TypeList<EnthalpyValue>

10.387.3 Constructor & Destructor Documentation

10.387.3.1 PressureForce() [1/2]

template<eos::models::PressureForceEOS EOST>
serif::equations::PressureForce< EOST >::PressureForce (
EOST equation_of_state,
const double enthalpy_power) [inline], [explicit]

The effective power describes P(h), not rho(h), for quadrature selection.

10.387.3.2 PressureForce() [2/2]

template<eos::models::PressureForceEOS EOST>
serif::equations::PressureForce< EOST >::PressureForce (
EOST equation_of_state) [inline], [explicit]

10.387.4 Member Function Documentation

10.387.4.1 additional_order()

template<eos::models::PressureForceEOS EOST>
int serif::equations::PressureForce< EOST >::additional_order (
EnthalpyValue ,
int) const [inline], [nodiscard]

10.387.4.2 evaluate()

template<eos::models::PressureForceEOS EOST>
PointLinearization< Operands > serif::equations::PressureForce< EOST >::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

The test divergence is scalar, so the physics supplies -P and -dP/dh. Test mapping and volume variation belong to the shared operator backend.

10.387.5 Member Data Documentation

10.387.5.1 m_enthalpy_power

template<eos::models::PressureForceEOS EOST>
double serif::equations::PressureForce< EOST >::m_enthalpy_power [private]

10.387.5.2 m_eos

template<eos::models::PressureForceEOS EOST>
EOST serif::equations::PressureForce< EOST >::m_eos [private]

10.387.5.3 name

template<eos::models::PressureForceEOS EOST>
std::string_view serif::equations::PressureForce< EOST >::name = "pressure force" [static], [constexpr]

The documentation for this class was generated from the following file:

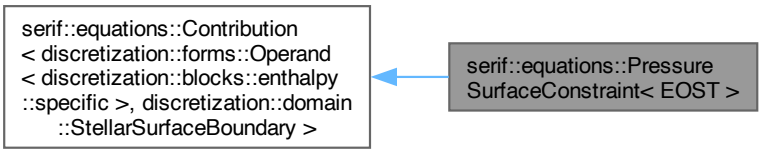
- src/include/serif/equations/pressure_force.hpp

10.388 serif::equations::PressureSurfaceConstraint< EOST > Class Template Reference

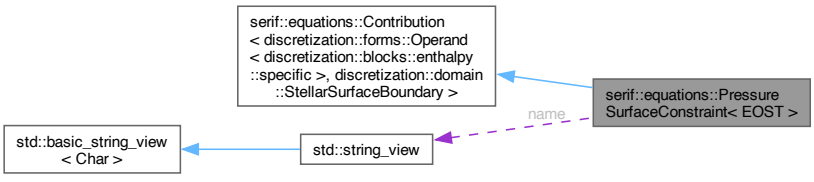
Prescribed pressure on the stellar boundary: h - h_EOS(P_surface) = 0.

#include <pressure_surface.hpp>

Inheritance diagram for serif::equations::PressureSurfaceConstraint< EOST >:



Collaboration diagram for serif::equations::PressureSurfaceConstraint< EOST >:



Public Types

- using Enthalpy = discretization::blocks::enthalpy::specific
- using Operands = utils::types::TypeList<discretization::forms::Operand<Enthalpy>>

Public Types inherited from serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::enthalpy::specific >

- using Test
- using TestOperand
- using Support

Public Member Functions

- PressureSurfaceConstraint (const EOST &equation_of_state, dimensions::PressureValue target_pressure)

  Resolve the fixed pressure through the EOS once; retain no borrowed EOS or geometry.
- PointLinearization< Operands > evaluate (const EquationPoint< Operands > &point) const

  This replaces a boundary row; it is not a surface or volume integral.
- dimensions::PressureValue target_pressure () const noexcept

Static Public Attributes

- static constexpr std::string_view name = "pressure surface constraint"
- static constexpr bool position_independent = true

Private Attributes

- dimensions::PressureValue m_target_pressure
- double m_target_enthalpy

10.388.1 Detailed Description

template<eos::models::EOSModel EOST>
requires eos::models::EOSSupportingRelation<EOST, eos::relations::SpecificEnthalpyFromPressure>
class serif::equations::PressureSurfaceConstraint< EOST >

Prescribed pressure on the stellar boundary: h - h_EOS(P_surface) = 0.

10.388.2 Member Typedef Documentation

10.388.2.1 Enthalpy

template<eos::models::EOSModel EOST>
using serif::equations::PressureSurfaceConstraint< EOST >::Enthalpy = discretization::blocks::enthalpy::specific

10.388.2.2 Operands

template<eos::models::EOSModel EOST>
using serif::equations::PressureSurfaceConstraint< EOST >::Operands = utils::types::TypeList<discretization::forms::Operand<Enthalpy>>

10.388.3 Constructor & Destructor Documentation

10.388.3.1 PressureSurfaceConstraint()

template<eos::models::EOSModel EOST>
serif::equations::PressureSurfaceConstraint< EOST >::PressureSurfaceConstraint (
const EOST & equation_of_state,
dimensions::PressureValue target_pressure) [inline]

Resolve the fixed pressure through the EOS once; retain no borrowed EOS or geometry.

10.388.4 Member Function Documentation

10.388.4.1 evaluate()

template<eos::models::EOSModel EOST>
PointLinearization< Operands > serif::equations::PressureSurfaceConstraint< EOST >::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

This replaces a boundary row; it is not a surface or volume integral.

10.388.4.2 target_pressure()

template<eos::models::EOSModel EOST>
dimensions::PressureValue serif::equations::PressureSurfaceConstraint< EOST >::target_pressure () const [inline], [nodiscard], [noexcept]

10.388.5 Member Data Documentation

10.388.5.1 m_target_enthalpy

template<eos::models::EOSModel EOST>
double serif::equations::PressureSurfaceConstraint< EOST >::m_target_enthalpy [private]

10.388.5.2 m_target_pressure

template<eos::models::EOSModel EOST>
dimensions::PressureValue serif::equations::PressureSurfaceConstraint< EOST >::m_target_pressure [private]

10.388.5.3 name

template<eos::models::EOSModel EOST>
std::string_view serif::equations::PressureSurfaceConstraint< EOST >::name = "pressure surface constraint" [static], [constexpr]

10.388.5.4 position_independent

template<eos::models::EOSModel EOST>
bool serif::equations::PressureSurfaceConstraint< EOST >::position_independent = true [static], [constexpr]

The documentation for this class was generated from the following file:

- src/include/serif/equations/pressure_surface.hpp

10.389 serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier > Struct Template Reference

Common physical metadata; numerical values and preparation belong to the dependency context and cache.

#include <compilation.hpp>

Public Types

- using EquationOfState = std::remove_cvref_t<EOS>
- using StateBindings = std::remove_cvref_t<Bindings>
- using Relation = RelationT
- using PhysicalCondition = IsobaricSurface
- using PhysicalQuantity = dimensions::Pressure
- using ResidualQuantity = std::conditional_t<ResolveCarrier, RowQuantity, dimensions::Pressure>
- using CarrierQuantity = RowQuantity
- using CarrierTerm = utils::dependencies::bound_type_t<StateBindings, RowQuantity>
- using Test = CarrierTerm
- using Support = discretization::domain::StellarSurfaceBoundary
- using EOSStateQuantities = std::conditional_t<ResolveCarrier, typename RelationQuantities<Relation>::NonPressureInputs, typename RelationQuantities<Relation>::AllInputs>
- using StateQuantities = utils::dependencies::CombinedKeyList<utils::dependencies::KeyList<RowQuantity>, EOSStateQuantities>
- using StateTerms = BoundTerms<StateBindings, StateQuantities>
- using EOSStateTerms = BoundTerms<StateBindings, EOSStateQuantities>
- using Operands = typename SurfaceOperands<StateTerms>::Type

Static Public Attributes

- static constexpr bool resolves_carrier = ResolveCarrier

10.389.1 Detailed Description

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool Resolve↩
Carrier>
struct serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >

Common physical metadata; numerical values and preparation belong to the dependency context and cache.

10.389.2 Member Typedef Documentation

10.389.2.1 CarrierQuantity

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::CarrierQuantity = RowQuantity

10.389.2.2 CarrierTerm

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::CarrierTerm = utils::dependencies::bound_type_t<StateBindings, RowQuantity>

10.389.2.3 EOSStateQuantities

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >::EOSStateQuantities = std::conditional_t<ResolveCarrier, typename RelationQuantities<Relation>::NonPressureInputs, typename RelationQuantities<Relation>::AllInputs>

10.389.2.4 EOSStateTerms

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::EOSStateTerms = BoundTerms<StateBindings, EOSStateQuantities>

10.389.2.5 EquationOfState

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::EquationOfState = std::remove_cvref_t<EOS>

10.389.2.6 Operands

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::Operands = typename SurfaceOperands<StateTerms>::Type

10.389.2.7 PhysicalCondition

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::PhysicalCondition = IsobaricSurface

10.389.2.8 PhysicalQuantity

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::PhysicalQuantity = dimensions::Pressure

10.389.2.9 Relation

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::Relation = RelationT

10.389.2.10 ResidualQuantity

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::ResidualQuantity = std::conditional_t<ResolveCarrier, RowQuantity, dimensions::Pressure>

10.389.2.11 StateBindings

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::StateBindings = std::remove_cvref_t<Bindings>

10.389.2.12 StateQuantities

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::StateQuantities = utils::dependencies::CombinedKeyList<utils::dependencies::KeyList<RowQuantity>, EOSStateQuantities>

10.389.2.13 StateTerms

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::StateTerms = BoundTerms<StateBindings, StateQuantities>

10.389.2.14 Support

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::Support = discretization::domain::StellarSurfaceBoundary

10.389.2.15 Test

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
using serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::Test = CarrierTerm

10.389.3 Member Data Documentation

10.389.3.1 resolves_carrier

template<typename EOS, typename Bindings, typename RowQuantity, typename RelationT, bool ResolveCarrier>
bool serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >↩
::resolves_carrier = ResolveCarrier [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/compilation.hpp

10.390 serif::geometry::surface::PressureSurfaceDescriptor Struct Reference

#include <isobaric.hpp>

Public Attributes

- double target_pressure

10.390.1 Member Data Documentation

10.390.1.1 target_pressure

double serif::geometry::surface::PressureSurfaceDescriptor::target_pressure

The documentation for this struct was generated from the following file:

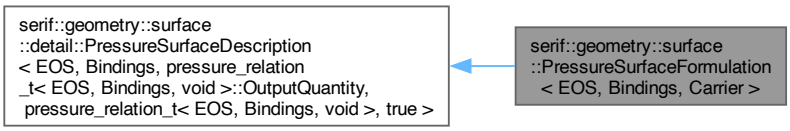
- src/include/serif/geometry/surface/isobaric.hpp

10.391 serif::geometry::surface::PressureSurfaceFormulation< EOS, Bindings, Carrier > Struct Template Reference

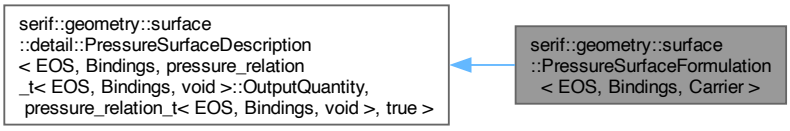
The boundary residual is carrier minus the EOS value at prescribed pressure; no carrier quantity is privileged.

#include <compilation.hpp>

Inheritance diagram for serif::geometry::surface::PressureSurfaceFormulation< EOS, Bindings, Carrier >:



Collaboration diagram for serif::geometry::surface::PressureSurfaceFormulation< EOS, Bindings, Carrier >:



Additional Inherited Members

Public Types inherited from serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, pressure_relation_t< EOS,

- using EquationOfState
- using StateBindings
- using Relation
- using PhysicalCondition
- using PhysicalQuantity
- using ResidualQuantity
- using CarrierQuantity
- using CarrierTerm
- using Test
- using Support
- using EOSStateQuantities
- using StateQuantities
- using StateTerms
- using EOSStateTerms
- using Operands

Static Public Attributes inherited from serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, pressure_relation_t< EOS,

- static constexpr bool resolves_carrier

10.391.1 Detailed Description

template<typename EOS, typename Bindings, typename Carrier = void>
requires PressureSurfaceCompilable<EOS, Bindings, Carrier>
struct serif::geometry::surface::PressureSurfaceFormulation< EOS, Bindings, Carrier >

The boundary residual is carrier minus the EOS value at prescribed pressure; no carrier quantity is privileged.

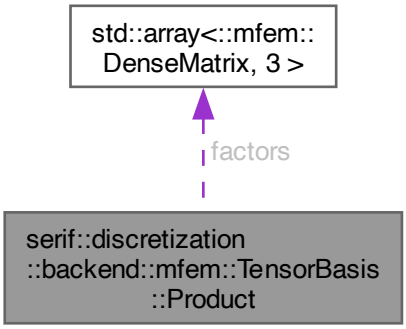
The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/compilation.hpp

10.392 serif::discretization::backend::mfem::TensorBasis::Product Struct Reference

A struct that represents a product of one-dimensional basis functions.

Collaboration diagram for serif::discretization::backend::mfem::TensorBasis::Product:



Public Attributes

- std::array<::mfem::DenseMatrix, 3 > factors
  The one-dimensional basis functions for each axis.
- ::mfem::Array< int > dofs
  The dofs associated with the product.
- int output
  The output index for the product (used for mapping to the correct component of the output vector)

10.392.1 Detailed Description

A struct that represents a product of one-dimensional basis functions.

10.392.2 Member Data Documentation

10.392.2.1 dofs

::mfem::Array<int> serif::discretization::backend::mfem::TensorBasis::Product::dofs

The dofs associated with the product.

10.392.2.2 factors

std::array<::mfem::DenseMatrix, 3> serif::discretization::backend::mfem::TensorBasis::Product::factors

The one-dimensional basis functions for each axis.

10.392.2.3 output

int serif::discretization::backend::mfem::TensorBasis::Product::output

The output index for the product (used for mapping to the correct component of the output vector)

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/backend/mfem/tensor_basis.hpp

10.393 serif::utils::dependencies::ProjectedSnapshot< FunctionT > Class Template Reference

#include <snapshot.hpp>

Public Member Functions

- ProjectedSnapshot (FunctionT function)
  Own the policy callable so its configuration travels with the source.
- template<typename T>
  requires std::invocable<const FunctionT&, const T&>
  auto capture (const T &value) const
  Capture a comparable value such as a resource revision or a tuple of revisions.
- template<typename CapturedT, typename T>
  requires requires(const CapturedT& captured, const ProjectedSnapshot& policy, const T& value) { { captured == policy.capture(value) } -> std::convertible_to<bool>; }
  bool matches (const CapturedT &captured, const T &value) const
  Recompute the tracking value without copying the resource itself.

Private Attributes

- FunctionT m_function

10.393.1 Constructor & Destructor Documentation

10.393.1.1 ProjectedSnapshot()

template<typename FunctionT>
serif::utils::dependencies::ProjectedSnapshot< FunctionT >::ProjectedSnapshot (
FunctionT function) [inline], [explicit]

Own the policy callable so its configuration travels with the source.

10.393.2 Member Function Documentation

10.393.2.1 capture()

```
template<typename FunctionT>
template<typename T>
requires std::invocable<const FunctionT&, const T&>
auto serif::utils::dependencies::ProjectedSnapshot< FunctionT >::capture (
                const T & value) const    [inline], [nodiscard]
```

Capture a comparable value such as a resource revision or a tuple of revisions.

10.393.2.2 matches()

```
template<typename FunctionT>
template<typename CapturedT, typename T>
requires requires(const CapturedT& captured, const ProjectedSnapshot& policy, const T& value) { { captured == policy.↩
capture(value) } -> std::convertible_to<bool>; }
bool serif::utils::dependencies::ProjectedSnapshot< FunctionT >::matches (
                const CapturedT & captured,
                const T & value) const    [inline], [nodiscard]
```

Recompute the tracking value without copying the resource itself.

10.393.3 Member Data Documentation

10.393.3.1 m_function

```
template<typename FunctionT>
FunctionT serif::utils::dependencies::ProjectedSnapshot< FunctionT >::m_function    [private]
```

The documentation for this class was generated from the following file:

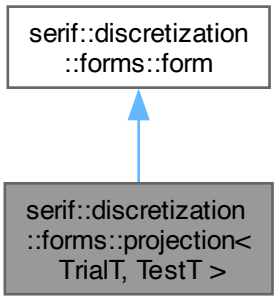
- src/include/serif/utils/dependencies/snapshot.hpp

10.394 serif::discretization::forms::projection< TrialT, TestT > Struct Template Reference

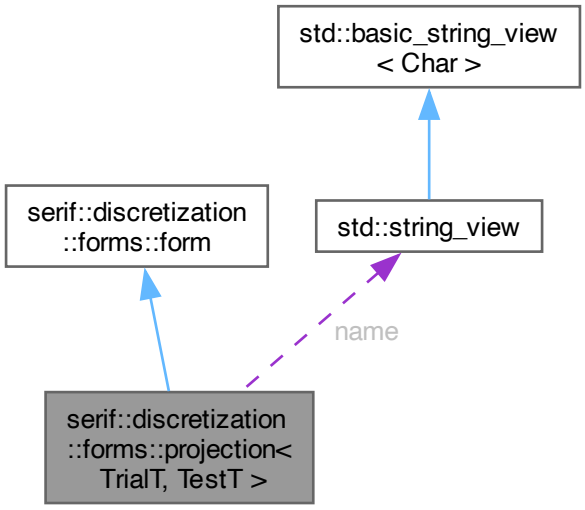
The projection form represents the projection operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the projection operation.

#include <forms.hpp>

Inheritance diagram for serif::discretization::forms::projection< TrialT, TestT >:



Collaboration diagram for serif::discretization::forms::projection< TrialT, TestT >:



Public Types

- using trial_type = TrialT

  The trial term type for the projection form.
- using test_type = TestT

  The test term type for the projection form.

- using operand_types = utils::types::TypeList<Operand<TrialT>, Operand<TestT>>

  Type list of operand types for the projection form, used for compile-time reflection and iteration over the operands.

Static Public Attributes

- static constexpr std::string_view name = meta::template_name<projection>()

  Compile time reflected name of the form.

10.394.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::projection< TrialT, TestT >

The projection form represents the projection operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the projection operation.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

10.394.2 Member Typedef Documentation

10.394.2.1 operand_types

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::projection< TrialT, TestT >::operand_types = utils::types::TypeList<Operand<TrialT>, Operand<TestT>>

Type list of operand types for the projection form, used for compile-time reflection and iteration over the operands.

10.394.2.2 test_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::projection< TrialT, TestT >::test_type = TestT

The test term type for the projection form.

10.394.2.3 trial_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::projection< TrialT, TestT >::trial_type = TrialT

The trial term type for the projection form.

10.394.3 Member Data Documentation

10.394.3.1 name

template<blocks::Term TrialT, blocks::Term TestT>
std::string_view serif::discretization::forms::projection< TrialT, TestT >::name = meta::template_name<projection>()
[static], [constexpr]

Compile time reflected name of the form.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/forms.hpp

10.395 serif::operators::backend::mfem::volume_keys::Quadrature↩ Linearization< Contribution > Struct Template Reference

#include <volume_data.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.396 serif::discretization::quadrature::QuadratureManifestOptions Struct Reference

#include <rules.hpp>

Public Attributes

- bool enabled = false
- bool include_repeated_queries = false
- std::optional< std::string > output_file

10.396.1 Member Data Documentation

10.396.1.1 enabled

bool serif::discretization::quadrature::QuadratureManifestOptions::enabled = false

10.396.1.2 include_repeated_queries

bool serif::discretization::quadrature::QuadratureManifestOptions::include_repeated_queries = false

10.396.1.3 output_file

std::optional<std::string> serif::discretization::quadrature::QuadratureManifestOptions::output_file

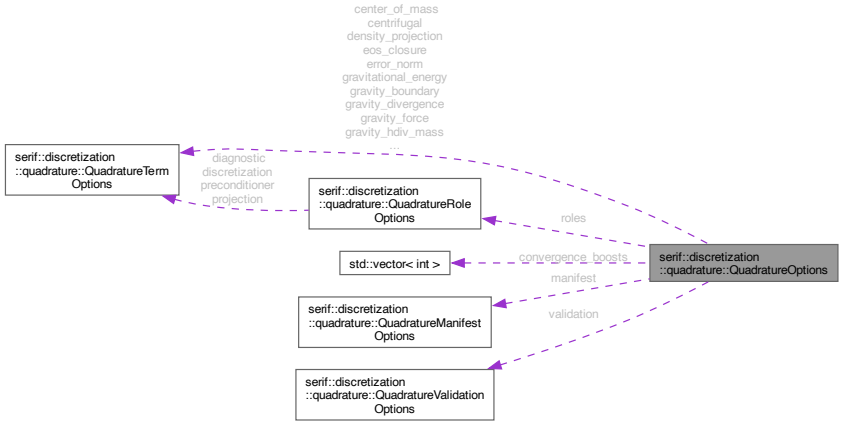
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

10.397 serif::discretization::quadrature::QuadratureOptions Struct Reference

#include <rules.hpp>

Collaboration diagram for serif::discretization::quadrature::QuadratureOptions:



Public Attributes

- Mode mode = Mode::production
- int global_boost = 0
- std::optional< int > fallback_fixed_order
- QuadratureTermOptions gravity_hdiv_mass
- QuadratureTermOptions gravity_divergence
- QuadratureTermOptions gravity_source
- QuadratureTermOptions gravity_force
- QuadratureTermOptions gravity_boundary

- QuadratureTermOptions centrifugal
- QuadratureTermOptions density_projection
- QuadratureTermOptions eos_closure
- QuadratureTermOptions hydrostatic_equilibrium
- QuadratureTermOptions isobaric_surface
- QuadratureTermOptions mesh_extension
- QuadratureTermOptions mass_conservation
- QuadratureTermOptions mass_normalization
- QuadratureTermOptions center_of_mass
- QuadratureTermOptions quadrupole
- QuadratureTermOptions gravitational_energy
- QuadratureTermOptions pressure_integral
- QuadratureTermOptions pressure_force
- QuadratureTermOptions virial
- QuadratureTermOptions error_norm
- QuadratureRoleOptions roles
- std::vector< int > convergence_boosts = {0, 2, 4}
- QuadratureManifestOptions manifest
- QuadratureValidationOptions validation

10.397.1 Member Data Documentation

10.397.1.1 center_of_mass

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::center_of_mass

10.397.1.2 centrifugal

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::centrifugal

10.397.1.3 convergence_boosts

std::vector<int> serif::discretization::quadrature::QuadratureOptions::convergence_boosts = {0, 2, 4}

10.397.1.4 density_projection

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::density_projection

10.397.1.5 eos_closure

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::eos_closure

10.397.1.6 error_norm

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::error_norm

10.397.1.7 fallback_fixed_order

std::optional<int> serif::discretization::quadrature::QuadratureOptions::fallback_fixed_order

10.397.1.8 global_boost

int serif::discretization::quadrature::QuadratureOptions::global_boost = 0

10.397.1.9 gravitational_energy

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::gravitational_energy

10.397.1.10 gravity_boundary

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::gravity_boundary

10.397.1.11 gravity_divergence

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::gravity_divergence

10.397.1.12 gravity_force

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::gravity_force

10.397.1.13 gravity_hdiv_mass

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::gravity_hdiv_mass

10.397.1.14 gravity_source

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::gravity_source

10.397.1.15 hydrostatic_equilibrium

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::hydrostatic_equilibrium

10.397.1.16 isobaric_surface

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::isobaric_surface

10.397.1.17 manifest

QuadratureManifestOptions serif::discretization::quadrature::QuadratureOptions::manifest

10.397.1.18 mass_conservation

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::mass_conservation

10.397.1.19 mass_normalization

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::mass_normalization

10.397.1.20 mesh_extension

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::mesh_extension

10.397.1.21 mode

Mode serif::discretization::quadrature::QuadratureOptions::mode = Mode::production

10.397.1.22 pressure_force

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::pressure_force

10.397.1.23 pressure_integral

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::pressure_integral

10.397.1.24 quadrupole

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::quadrupole

10.397.1.25 roles

QuadratureRoleOptions serif::discretization::quadrature::QuadratureOptions::roles

10.397.1.26 validation

QuadratureValidationOptions serif::discretization::quadrature::QuadratureOptions::validation

10.397.1.27 virial

QuadratureTermOptions serif::discretization::quadrature::QuadratureOptions::virial

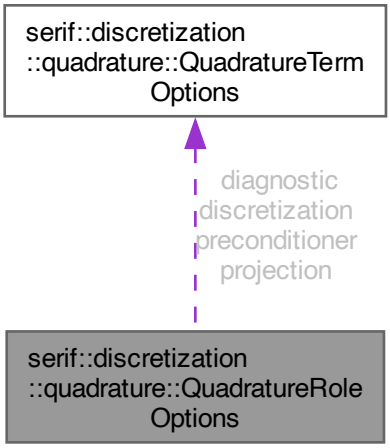
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

10.398 serif::discretization::quadrature::QuadratureRoleOptions Struct Reference

#include <rules.hpp>

Collaboration diagram for serif::discretization::quadrature::QuadratureRoleOptions:



Public Attributes

- QuadratureTermOptions discretization
- QuadratureTermOptions preconditioner
- QuadratureTermOptions diagnostic
- QuadratureTermOptions projection

10.398.1 Member Data Documentation

10.398.1.1 diagnostic

QuadratureTermOptions serif::discretization::quadrature::QuadratureRoleOptions::diagnostic

10.398.1.2 discretization

QuadratureTermOptions serif::discretization::quadrature::QuadratureRoleOptions::discretization

10.398.1.3 preconditioner

QuadratureTermOptions serif::discretization::quadrature::QuadratureRoleOptions::preconditioner

10.398.1.4 projection

QuadratureTermOptions serif::discretization::quadrature::QuadratureRoleOptions::projection

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

10.399 serif::discretization::quadrature::QuadratureTermOptions Struct Reference

#include <rules.hpp>

Public Attributes

- std::optional< int > fixed_order
- int additional_boost = 0

10.399.1 Member Data Documentation

10.399.1.1 additional_boost

int serif::discretization::quadrature::QuadratureTermOptions::additional_boost = 0

10.399.1.2 fixed_order

std::optional<int> serif::discretization::quadrature::QuadratureTermOptions::fixed_order

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

10.400 serif::discretization::quadrature::QuadratureValidationOptions Struct Reference

#include <rules.hpp>

Public Attributes

- bool require_explicit_base_order = false
- bool require_explicit_mfem_rule = false
- bool reject_negative_boosts = true
- bool report_unused_overrides = true

10.400.1 Member Data Documentation

10.400.1.1 reject_negative_boosts

bool serif::discretization::quadrature::QuadratureValidationOptions::reject_negative_boosts = true

10.400.1.2 report_unused_overrides

bool serif::discretization::quadrature::QuadratureValidationOptions::report_unused_overrides = true

10.400.1.3 require_explicit_base_order

bool serif::discretization::quadrature::QuadratureValidationOptions::require_explicit_base_order = false

10.400.1.4 require_explicit_mfem_rule

bool serif::discretization::quadrature::QuadratureValidationOptions::require_explicit_mfem_rule = false

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

10.401 serif::dimensions::QuantityAt< Index, Quantities > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/quantities.hpp

10.402 serif::dimensions::QuantityAt< Index, QuantityList< Quantities... > > Struct Template Reference

#include <quantities.hpp>

Public Types

- using Type = std::tuple_element_t<Index, std::tuple<Quantities...>>

10.402.1 Member Typedef Documentation

10.402.1.1 Type

template<std::size_t Index, typename... Quantities>
using serif::dimensions::QuantityAt< Index, QuantityList< Quantities... > >::Type = std::tuple_element_t<Index, std::tuple<Quantities...>>

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/quantities.hpp

10.403 serif::dimensions::QuantityList< Quantities > Struct Template Reference

#include <quantities.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/quantities.hpp

10.404 serif::dimensions::QuantityOf< Candidate > Struct Template Reference

#include <quantities.hpp>

10.404.1 Detailed Description

template<typename Candidate>
struct serif::dimensions::QuantityOf< Candidate >

Template Parameters

Candidate	Some type that may be able to report what its dimension is

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/quantities.hpp

10.405 serif::dimensions::QuantityOf< QuantityValue< Quantity > > Struct Template Reference

#include <quantities.hpp>

Public Types

- using Type = Quantity

10.405.1 Member Typedef Documentation

10.405.1.1 Type

template<PhysicalQuantityType Quantity>
using serif::dimensions::QuantityOf< QuantityValue< Quantity > >::Type = Quantity

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/quantities.hpp

10.406 serif::dimensions::QuantityValue< Quantity > Class Template Reference

This is the non derivative equivalent to PartialDerivative. This associates a dimension with a value and, if that value can be determined at compile time, can be decomposed to a transparent double through .value()

#include <quantities.hpp>

Public Member Functions

- constexpr QuantityValue (const double value) noexcept
- constexpr double value () const noexcept

Private Attributes

- double m_value

  Numeric value associated with the dimension of QuantityValue.

Friends

- constexpr bool operator== (const QuantityValue &, const QuantityValue &) noexcept=default
- constexpr QuantityValue operator+ (const QuantityValue &lhs, const QuantityValue &rhs) noexcept
- constexpr QuantityValue operator- (const QuantityValue &lhs, const QuantityValue &rhs) noexcept
- template<IsNumeric Scalar>
  constexpr QuantityValue operator* (const QuantityValue &lhs, const Scalar rhs) noexcept
- template<IsNumeric Scalar>
  constexpr QuantityValue operator* (const Scalar lhs, const QuantityValue &rhs) noexcept
- template<IsNumeric Scalar>
  constexpr QuantityValue operator/ (const QuantityValue &lhs, const Scalar rhs) noexcept
- template<IsNumeric Scalar>
  constexpr std::partial_ordering operator<=> (const QuantityValue &lhs, const Scalar rhs) noexcept
- template<IsNumeric Scalar>
  constexpr std::partial_ordering operator<=> (const Scalar lhs, const QuantityValue &rhs) noexcept
- constexpr std::partial_ordering operator<=> (const QuantityValue &lhs, const QuantityValue &rhs) noexcept

10.406.1 Detailed Description

template<PhysicalQuantityType Quantity>
class serif::dimensions::QuantityValue< Quantity >

This is the non derivative equivalent to PartialDerivative. This associates a dimension with a value and, if that value can be determined at compile time, can be decomposed to a transparent double through .value()

Template Parameters

Quantity	Some valid PhysicalQuantityType. Note all the dimensions in dimensions.hpp are of type PhysicalQuantity. There should be no other dimensions in the code.

10.406.2 Constructor & Destructor Documentation

10.406.2.1 QuantityValue()

template<PhysicalQuantityType Quantity>
serif::dimensions::QuantityValue< Quantity >::QuantityValue (
const double value) [inline], [explicit], [constexpr], [noexcept]

10.406.3 Member Function Documentation

10.406.3.1 value()

```
template<PhysicalQuantityType Quantity>
double serif::dimensions::QuantityValue< Quantity >::value () const    [inline], [nodiscard], [constexpr], [noexcept]
```

10.406.4 Friends And Related Symbol Documentation

10.406.4.1 operator∗ [1/2]

```
template<PhysicalQuantityType Quantity>
template<IsNumeric Scalar>
QuantityValue operator∗ (
                const QuantityValue< Quantity > & lhs,
                const Scalar rhs)    [friend]
```

@breif Overload of the ∗ operator for a left quantity and some right numeric, scalar, value.

Template Parameters

Scalar	A value satisfying IsNumeric

Parameters

lhs	some quantity value
rhs	some scalar value

Returns

some quantity value with the same type as lhs but with value = lhs.m_value ∗ scalar

10.406.4.2 operator∗ [2/2]

```
template<PhysicalQuantityType Quantity>
template<IsNumeric Scalar>
QuantityValue operator∗ (
                const Scalar lhs,
                const QuantityValue< Quantity > & rhs)    [friend]
```

@breif Overload of the ∗ operator for a right quantity and some left numeric, scalar, value.

Template Parameters

Scalar	A value satisfying IsNumeric

Parameters

lhs	some scalar value
rhs	some quantity value

Returns

some quantity value with the same type as lhs but with value = scalar ∗ lhs.m_value

10.406.4.3 operator+

```
template<PhysicalQuantityType Quantity>
QuantityValue operator+ (
                const QuantityValue< Quantity > & lhs,
                const QuantityValue< Quantity > & rhs)   [friend]
```

@breif overload of + operator for two quantities with the same PhysicalQuantityType

Parameters

lhs	One Quantity
rhs	Other Quantity

Returns

a new quantity with the same PhysicalQuantityType but with a value = lhs.m_value + rhs.m_value

10.406.4.4 operator-

```
template<PhysicalQuantityType Quantity>
QuantityValue operator- (
                const QuantityValue< Quantity > & lhs,
                const QuantityValue< Quantity > & rhs)   [friend]
```

@breif overload of - operator for two quantities with the same PhysicalQuantityType

Parameters

lhs	One Quantity
rhs	Other Quantity

Returns

a new quantity with the same PhysicalQuantityType but with a value = lhs.m_value - rhs.m_value

10.406.4.5 operator/

```
template<PhysicalQuantityType Quantity>
template<IsNumeric Scalar>
QuantityValue operator/ (
                const QuantityValue< Quantity > & lhs,
                const Scalar rhs)   [friend]
```

@breif Overload of the / operator for a left quantity and some right numeric, scalar, value.

Template Parameters

Scalar	A value satisfying IsNumeric

Parameters

lhs	some quantity value
rhs	some scalar value

Returns

some quantity value with the same type as lhs but with value = lhs.m_value / scalar

10.406.4.6 operator<=> [1/3]

```
template<PhysicalQuantityType Quantity>
std::partial_ordering operator<=> (
        const QuantityValue< Quantity > & lhs,
        const QuantityValue< Quantity > & rhs)    [friend]
```

@breif partial ordering overload to compare two quantity values of the same type

Parameters

lhs	Some quantity value
rhs	Other quantity value

Returns

the partial ordering of lhs.m_value <=> rhs.m_value

10.406.4.7 operator<=> [2/3]

```
template<PhysicalQuantityType Quantity>
template<IsNumeric Scalar>
std::partial_ordering operator<=> (
        const QuantityValue< Quantity > & lhs,
        const Scalar rhs)    [friend]
```

@breif Partial ordering overload for some left quantity and right scalar

Template Parameters

Scalar	A value satisfying IsNumeric

Parameters

lhs	some quantity value
rhs	some scalar value

Returns

The partial ordering comparing lhs.m_value to scalar

10.406.4.8 operator<=> [3/3]

```
template<PhysicalQuantityType Quantity>
template<IsNumeric Scalar>
std::partial_ordering operator<=> (
            const Scalar lhs,
            const QuantityValue< Quantity > & rhs)    [friend]
```

@breif Partial ordering overload for some right quantity and left scalar

Template Parameters

Scalar	A value satisfying IsNumeric

Parameters

lhs	some scalar value
rhs	some quantity value

Returns

The partial ordering comparing scalar to lhs.m_value

10.406.4.9 operator==

```
template<PhysicalQuantityType Quantity>
bool operator== (
            const QuantityValue< Quantity > & ,
            const QuantityValue< Quantity > & )    [friend]
```

@breif Default comparison operator for two QuantityValues. Note that there is an implicit requirement these be of the same specialization. E.g. each have the same PhysicalQuantityType Quantity type parameter. Because this is the default and this class has only one trivially comparable member this just checks if m_↩ value == m_value.

Returns

true if the values in both are equal.

10.406.5 Member Data Documentation

10.406.5.1 m_value

template<PhysicalQuantityType Quantity>
double serif::dimensions::QuantityValue< Quantity >::m_value [private]

Numeric value associated with the dimension of QuantityValue.

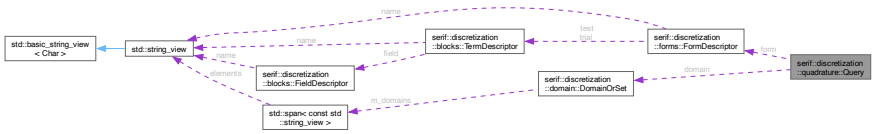
The documentation for this class was generated from the following file:

- src/include/serif/dimensions/quantities.hpp

10.407 serif::discretization::quadrature::Query Struct Reference

#include <query.hpp>

Collaboration diagram for serif::discretization::quadrature::Query:



Public Attributes

- forms::FormDescriptor form
- QuadratureRole role = QuadratureRole::discretization
- domain::DomainOrSet domain = domain::AllDomains{}
- MappingKind mapping = MappingKind::none
- int trial_order = 0
- int test_order = 0
- int coefficient_order = 0
- int geometry_weight_order = 0
- int trial_order_reduction = 0
- int test_order_reduction = 0
- std::optional< int > base_order

10.407.1 Member Data Documentation

10.407.1.1 base_order

std::optional<int> serif::discretization::quadrature::Query::base_order

10.407.1.2 coefficient_order

int serif::discretization::quadrature::Query::coefficient_order = 0

10.407.1.3 domain

domain::DomainOrSet serif::discretization::quadrature::Query::domain = domain::AllDomains{}

10.407.1.4 form

forms::FormDescriptor serif::discretization::quadrature::Query::form

10.407.1.5 geometry_weight_order

int serif::discretization::quadrature::Query::geometry_weight_order = 0

10.407.1.6 mapping

MappingKind serif::discretization::quadrature::Query::mapping = MappingKind::none

10.407.1.7 role

QuadratureRole serif::discretization::quadrature::Query::role = QuadratureRole::discretization

10.407.1.8 test_order

int serif::discretization::quadrature::Query::test_order = 0

10.407.1.9 test_order_reduction

int serif::discretization::quadrature::Query::test_order_reduction = 0

10.407.1.10 trial_order

int serif::discretization::quadrature::Query::trial_order = 0

10.407.1.11 trial_order_reduction

int serif::discretization::quadrature::Query::trial_order_reduction = 0

The documentation for this struct was generated from the following file:

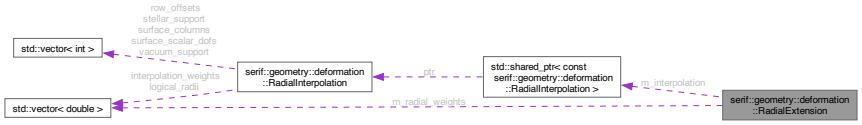
- src/include/serif/discretization/quadrature/query.hpp

10.408 serif::geometry::deformation::RadialExtension Class Reference

Extend one interleaved surface vector into the MFEM component-block volume layout. Stellar ownership wins at shared nodes. Both actions use the same scalar interpolation, so their pairing is Euclidean virtual work without an additional surface-area weight.

#include <radial_extension.hpp>

Collaboration diagram for serif::geometry::deformation::RadialExtension:



Public Types

- using InputSpace = spaces::SurfaceDisplacement
- using OutputSpace = spaces::VolumeDisplacement

Public Member Functions

- const std::shared_ptr< const RadialInterpolation > & interpolation () const noexcept

  Expose shared interpolation read-only for diagnostics and independent geometry checks.
- std::span< const double > radial_weights () const noexcept

  Expose the ownership-selected radial response without permitting mutation.
- std::size_t input_size () const noexcept

  Return the interleaved Cartesian surface-vector size.
- std::size_t output_size () const noexcept

  Return the Cartesian component-block volume-vector size.
- DeformationStatus apply (std::span< const double > surface_displacement, std::span< double > volume_displacement) const

  Apply the fused radial extension without allocating action scratch. Input and output must not overlap. Discard output if an action reports failure.
- DeformationStatus apply_transpose (std::span< const double > volume_force, std::span< double > surface_force) const

  Transfer volume forces to surface forces using exactly the forward ownership.

Static Public Attributes

- static constexpr bool is_linear = true

Private Member Functions

- RadialExtension (std::shared_ptr< const RadialInterpolation > interpolation, std::vector< double > radial_weights)

  Store already validated data without performing hidden numerical preparation.

Private Attributes

- std::shared_ptr< const RadialInterpolation > m_interpolation
- std::vector< double > m_radial_weights

Friends

- std::expected< RadialExtension, DeformationFailure > make_radial_extension (std::shared_ptr< const RadialInterpolation > interpolation, PowerLawRadialExtension interior={}, FixedInfinityRadialExtension vacuum={})

  Compile regional weights while sharing the immutable surface interpolation. Callers supplying a shared pointer must treat its data as immutable after compilation; the dependency graph publishes a const owning value for precisely this reason.

10.408.1 Detailed Description

Extend one interleaved surface vector into the MFEM component-block volume layout. Stellar ownership wins at shared nodes. Both actions use the same scalar interpolation, so their pairing is Euclidean virtual work without an additional surface-area weight.

10.408.2 Member Typedef Documentation

10.408.2.1 InputSpace

using serif::geometry::deformation::RadialExtension::InputSpace = spaces::SurfaceDisplacement

10.408.2.2 OutputSpace

using serif::geometry::deformation::RadialExtension::OutputSpace = spaces::VolumeDisplacement

10.408.3 Constructor & Destructor Documentation

10.408.3.1 RadialExtension()

serif::geometry::deformation::RadialExtension::RadialExtension (
std::shared_ptr< const RadialInterpolation > interpolation,
std::vector< double > radial_weights) [private]

Store already validated data without performing hidden numerical preparation.

10.408.4 Member Function Documentation

10.408.4.1 apply()

DeformationStatus serif::geometry::deformation::RadialExtension::apply (
std::span< const double > surface_displacement,
std::span< double > volume_displacement) const [nodiscard]

Apply the fused radial extension without allocating action scratch. Input and output must not overlap. Discard output if an action reports failure.

10.408.4.2 apply_transpose()

DeformationStatus serif::geometry::deformation::RadialExtension::apply_transpose (
std::span< const double > volume_force,
std::span< double > surface_force) const [nodiscard]

Transfer volume forces to surface forces using exactly the forward ownership.

10.408.4.3 input_size()

std::size_t serif::geometry::deformation::RadialExtension::input_size () const [inline], [nodiscard], [noexcept]

Return the interleaved Cartesian surface-vector size.

10.408.4.4 interpolation()

const std::shared_ptr< const RadialInterpolation > & serif::geometry::deformation::RadialExtension::interpolation () const [inline], [nodiscard], [noexcept]

Expose shared interpolation read-only for diagnostics and independent geometry checks.

10.408.4.5 output_size()

std::size_t serif::geometry::deformation::RadialExtension::output_size () const [inline], [nodiscard], [noexcept]

Return the Cartesian component-block volume-vector size.

10.408.4.6 radial_weights()

std::span< const double > serif::geometry::deformation::RadialExtension::radial_weights () const [inline], [nodiscard], [noexcept]

Expose the ownership-selected radial response without permitting mutation.

10.408.5 Friends And Related Symbol Documentation

10.408.5.1 make_radial_extension

std::expected< RadialExtension, DeformationFailure > make_radial_extension (
std::shared_ptr< const RadialInterpolation > interpolation,
PowerLawRadialExtension interior = {},
FixedInfinityRadialExtension vacuum = {}) [friend]

Compile regional weights while sharing the immutable surface interpolation. Callers supplying a shared pointer must treat its data as immutable after compilation; the dependency graph publishes a const owning value for precisely this reason.

10.408.6 Member Data Documentation

10.408.6.1 is_linear

bool serif::geometry::deformation::RadialExtension::is_linear = true [static], [constexpr]

10.408.6.2 m_interpolation

std::shared_ptr<const RadialInterpolation> serif::geometry::deformation::RadialExtension::m_interpolation [private]

10.408.6.3 m_radial_weights

std::vector<double> serif::geometry::deformation::RadialExtension::m_radial_weights [private]

The documentation for this class was generated from the following files:

- src/include/serif/geometry/deformation/radial_extension.hpp
- src/lib/serif/geometry/deformation/radial_extension.cpp

10.409 serif::mapping::compactification::KelvinCompactification::RadialFactors Struct Reference

Public Attributes

- double compactification_coordinate
- double computational_radius
- double scale
- double scale_derivative

10.409.1 Member Data Documentation

10.409.1.1 compactification_coordinate

double serif::mapping::compactification::KelvinCompactification::RadialFactors::compactification_coordinate

10.409.1.2 computational_radius

double serif::mapping::compactification::KelvinCompactification::RadialFactors::computational_radius

10.409.1.3 scale

double serif::mapping::compactification::KelvinCompactification::RadialFactors::scale

10.409.1.4 scale_derivative

double serif::mapping::compactification::KelvinCompactification::RadialFactors::scale_derivative

The documentation for this struct was generated from the following file:

- src/include/serif/mapping/compactification/kelvin.hpp

10.410 serif::geometry::deformation::keys::RadialInterpolation Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

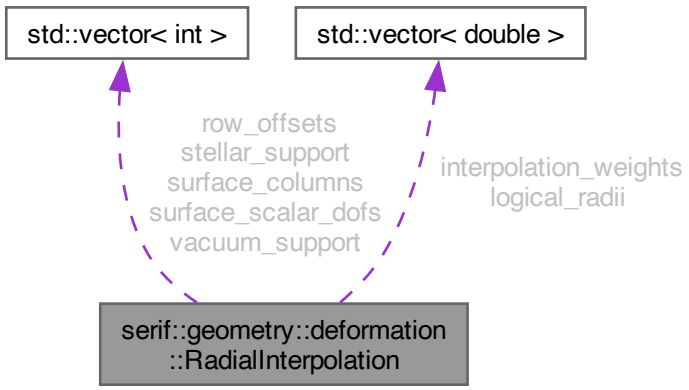
- src/include/serif/geometry/deformation/computations.hpp

10.411 serif::geometry::deformation::RadialInterpolation Struct Reference

Own the surface interpolation shared by the stellar and vacuum extensions.

#include <radial_extension.hpp>

Collaboration diagram for serif::geometry::deformation::RadialInterpolation:



Public Member Functions

- std::size_t scalar_dof_count () const noexcept
  Return the volume scalar-node count, independent of vector layout.
- std::size_t surface_node_count () const noexcept
  Return the canonical number of stellar surface nodes.

Public Attributes

- std::vector< int > surface_scalar_dofs
- std::vector< int > row_offsets
- std::vector< int > surface_columns
- std::vector< double > interpolation_weights
- std::vector< double > logical_radii
- std::vector< int > stellar_support
- std::vector< int > vacuum_support
- double stellar_radius {}
- double infinity_radius {}

10.411.1 Detailed Description

Own the surface interpolation shared by the stellar and vacuum extensions.

10.411.2 Member Function Documentation

10.411.2.1 scalar_dof_count()

std::size_t serif::geometry::deformation::RadialInterpolation::scalar_dof_count () const [inline], [nodiscard], [noexcept]

Return the volume scalar-node count, independent of vector layout.

10.411.2.2 surface_node_count()

std::size_t serif::geometry::deformation::RadialInterpolation::surface_node_count () const [inline], [nodiscard], [noexcept]

Return the canonical number of stellar surface nodes.

10.411.3 Member Data Documentation

10.411.3.1 infinity_radius

double serif::geometry::deformation::RadialInterpolation::infinity_radius {}

10.411.3.2 interpolation_weights

std::vector<double> serif::geometry::deformation::RadialInterpolation::interpolation_weights

10.411.3.3 logical_radii

std::vector<double> serif::geometry::deformation::RadialInterpolation::logical_radii

10.411.3.4 row_offsets

std::vector<int> serif::geometry::deformation::RadialInterpolation::row_offsets

10.411.3.5 stellar_radius

double serif::geometry::deformation::RadialInterpolation::stellar_radius {}

10.411.3.6 stellar_support

std::vector<int> serif::geometry::deformation::RadialInterpolation::stellar_support

10.411.3.7 surface_columns

std::vector<int> serif::geometry::deformation::RadialInterpolation::surface_columns

10.411.3.8 surface_scalar_dofs

std::vector<int> serif::geometry::deformation::RadialInterpolation::surface_scalar_dofs

10.411.3.9 vacuum_support

std::vector<int> serif::geometry::deformation::RadialInterpolation::vacuum_support

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/radial_extension.hpp

10.412 serif::discretization::functional_space::Rank< R > Struct Template Reference

Compile-time descriptor for the rank of a functional space.

#include <space.hpp>

Static Public Attributes

- static constexpr std::size_t value = R
  The rank of the functional space.

10.412.1 Detailed Description

template<std::size_t R>
struct serif::discretization::functional_space::Rank< R >

Compile-time descriptor for the rank of a functional space.

Template Parameters

R	The rank of the functional space.

10.412.2 Member Data Documentation

10.412.2.1 value

template<std::size_t R>
std::size_t serif::discretization::functional_space::Rank< R >::value = R [static], [constexpr]

The rank of the functional space.

The documentation for this struct was generated from the following file:

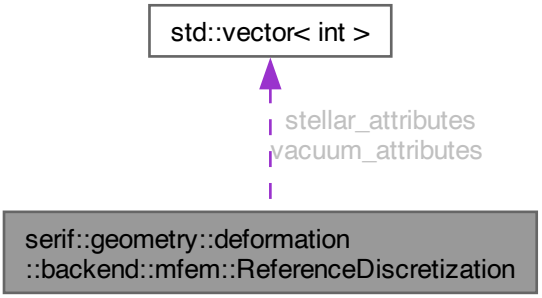
- src/include/serif/discretization/functional_space/space.hpp

10.413 serif::geometry::deformation::backend::mfem::ReferenceDiscretization Struct Reference

Borrow the paired reference meshes; attributes retain their physical domain meaning.

#include <inputs.hpp>

Collaboration diagram for serif::geometry::deformation::backend::mfem::ReferenceDiscretization:



Public Attributes

- const ::mfem::FiniteElementSpace & scalar_space
- const ::mfem::FiniteElementSpace & displacement_space
- ::mfem::Mesh & logical_mesh
- std::vector< int > stellar_attributes
- std::vector< int > vacuum_attributes
- int surface_attribute
- int infinity_attribute

10.413.1 Detailed Description

Borrow the paired reference meshes; attributes retain their physical domain meaning.

10.413.2 Member Data Documentation

10.413.2.1 displacement_space

const ::mfem::FiniteElementSpace& serif::geometry::deformation::backend::mfem::ReferenceDiscretization::displacement_space

10.413.2.2 infinity_attribute

int serif::geometry::deformation::backend::mfem::ReferenceDiscretization::infinity_attribute

10.413.2.3 logical_mesh

::mfem::Mesh& serif::geometry::deformation::backend::mfem::ReferenceDiscretization::logical_mesh

10.413.2.4 scalar_space

const ::mfem::FiniteElementSpace& serif::geometry::deformation::backend::mfem::ReferenceDiscretization::scalar_space

10.413.2.5 stellar_attributes

std::vector<int> serif::geometry::deformation::backend::mfem::ReferenceDiscretization::stellar_attributes

10.413.2.6 surface_attribute

int serif::geometry::deformation::backend::mfem::ReferenceDiscretization::surface_attribute

10.413.2.7 vacuum_attributes

std::vector<int> serif::geometry::deformation::backend::mfem::ReferenceDiscretization::vacuum_attributes

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/backend/mfem/inputs.hpp

10.414 serif::geometry::deformation::keys::ReferenceDiscretization Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

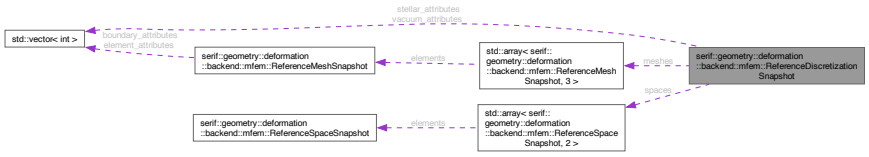
- src/include/serif/geometry/deformation/computations.hpp

10.415 serif::geometry::deformation::backend::mfem::Reference↩ DiscretizationSnapshot Struct Reference

One comparable snapshot covers all inputs used to compile fixed deformation data.

#include <inputs.hpp>

Collaboration diagram for serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot:



Public Member Functions

- bool operator== (const ReferenceDiscretizationSnapshot &) const =default

  Preserve identities, both reference coordinate revisions, and semantic selections.

Public Attributes

- std::array< ReferenceMeshSnapshot, 3 > meshes
- std::array< ReferenceSpaceSnapshot, 2 > spaces
- std::vector< int > stellar_attributes
- std::vector< int > vacuum_attributes
- int surface_attribute
- int infinity_attribute

10.415.1 Detailed Description

One comparable snapshot covers all inputs used to compile fixed deformation data.

10.415.2 Member Function Documentation

10.415.2.1 operator==()

bool serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot::operator== (
const ReferenceDiscretizationSnapshot &) const [default]

Preserve identities, both reference coordinate revisions, and semantic selections.

10.415.3 Member Data Documentation

10.415.3.1 infinity_attribute

int serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot::infinity_attribute

10.415.3.2 meshes

std::array<ReferenceMeshSnapshot, 3> serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot::meshes

10.415.3.3 spaces

std::array<ReferenceSpaceSnapshot, 2> serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot::spaces

10.415.3.4 stellar_attributes

std::vector<int> serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot::stellar_attributes

10.415.3.5 surface_attribute

int serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot::surface_attribute

10.415.3.6 vacuum_attributes

std::vector<int> serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot::vacuum_attributes

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/backend/mfem/inputs.hpp

10.416 serif::geometry::deformation::backend::mfem::ReferenceDiscretizationTracking Struct Reference

Let the generic dependency engine track MFEM metadata without owning meshes.

#include <inputs.hpp>

Public Member Functions

- ReferenceDiscretizationSnapshot capture (const ReferenceDiscretization &input) const
  Record all mutable reference metadata before any numerical graph stage runs.
- bool matches (const ReferenceDiscretizationSnapshot &snapshot, const ReferenceDiscretization &input) const
  Reuse fixed geometry only while its exact source metadata agrees.
- const void * identity (const ReferenceDiscretization &) const noexcept
  The snapshot contains persistent object identities; temporary wrappers have no identity.

10.416.1 Detailed Description

Let the generic dependency engine track MFEM metadata without owning meshes.

10.416.2 Member Function Documentation

10.416.2.1 capture()

ReferenceDiscretizationSnapshot serif::geometry::deformation::backend::mfem::ReferenceDiscretizationTracking::capture (
const ReferenceDiscretization & input) const [nodiscard]

Record all mutable reference metadata before any numerical graph stage runs.

10.416.2.2 identity()

const void * serif::geometry::deformation::backend::mfem::ReferenceDiscretizationTracking::identity (
const ReferenceDiscretization &) const [inline], [nodiscard], [noexcept]

The snapshot contains persistent object identities; temporary wrappers have no identity.

10.416.2.3 matches()

bool serif::geometry::deformation::backend::mfem::ReferenceDiscretizationTracking::matches (
const ReferenceDiscretizationSnapshot & snapshot,
const ReferenceDiscretization & input) const [nodiscard]

Reuse fixed geometry only while its exact source metadata agrees.

The documentation for this struct was generated from the following files:

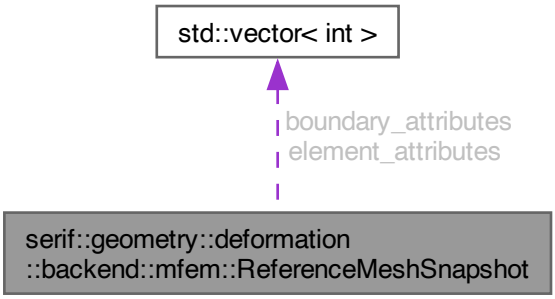
- src/include/serif/geometry/deformation/backend/mfem/inputs.hpp
- src/lib/serif/geometry/deformation/backend/mfem/inputs.cpp

10.417 serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot Struct Reference

Capture coordinate revisions as well as topology and mutable material/boundary labels.

#include <inputs.hpp>

Collaboration diagram for serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot:



Public Member Functions

- bool operator== (const ReferenceMeshSnapshot &) const =default

  Exact metadata equality avoids tolerance-based stale geometry reuse.

Public Attributes

- const ::mfem::Mesh * identity {}
- long sequence {-1}
- long nodes_sequence {-1}
- int vertices {}
- std::vector< int > element_attributes
- std::vector< int > boundary_attributes

10.417.1 Detailed Description

Capture coordinate revisions as well as topology and mutable material/boundary labels.

10.417.2 Member Function Documentation

10.417.2.1 operator==()

bool serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot::operator== (
const ReferenceMeshSnapshot &) const [default]

Exact metadata equality avoids tolerance-based stale geometry reuse.

10.417.3 Member Data Documentation

10.417.3.1 boundary_attributes

std::vector<int> serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot::boundary_attributes

10.417.3.2 element_attributes

std::vector<int> serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot::element_attributes

10.417.3.3 identity

const ::mfem::Mesh* serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot::identity {}

10.417.3.4 nodes_sequence

long serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot::nodes_sequence {-1}

10.417.3.5 sequence

long serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot::sequence {-1}

10.417.3.6 vertices

int serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot::vertices {}

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/backend/mfem/inputs.hpp

10.418 serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot Struct Reference

Track the coefficient interpretation independently of numerical coefficient values.

#include <inputs.hpp>

Public Member Functions

- bool operator== (const ReferenceSpaceSnapshot &) const =default
  A distinct finite-element binding is a distinct reference calculation.

Public Attributes

- const ::mfem::FiniteElementSpace * identity
- const ::mfem::FiniteElementCollection * collection
- long sequence
- int size
- int vector_dimension
- int ordering

10.418.1 Detailed Description

Track the coefficient interpretation independently of numerical coefficient values.

10.418.2 Member Function Documentation

10.418.2.1 operator==()

bool serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot::operator== (
const ReferenceSpaceSnapshot &) const [default]

A distinct finite-element binding is a distinct reference calculation.

10.418.3 Member Data Documentation

10.418.3.1 collection

const ::mfem::FiniteElementCollection* serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot::collection

10.418.3.2 identity

const ::mfem::FiniteElementSpace* serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot::identity

10.418.3.3 ordering

int serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot::ordering

10.418.3.4 sequence

long serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot::sequence

10.418.3.5 size

int serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot::size

10.418.3.6 vector_dimension

int serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot::vector_dimension

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/backend/mfem/inputs.hpp

10.419 serif::geometry::deformation::keys::ReferenceSurface Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

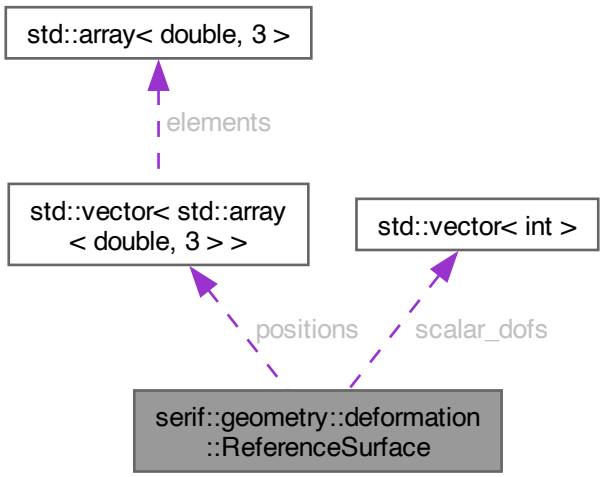
- src/include/serif/geometry/deformation/computations.hpp

10.420 serif::geometry::deformation::ReferenceSurface Struct Reference

Own fixed surface positions and their canonical ascending scalar-node indices.

#include <reference_surface.hpp>

Collaboration diagram for serif::geometry::deformation::ReferenceSurface:



Public Attributes

- std::vector< std::array< double, 3 > > positions
- std::vector< int > scalar_dofs

10.420.1 Detailed Description

Own fixed surface positions and their canonical ascending scalar-node indices.

10.420.2 Member Data Documentation

10.420.2.1 positions

std::vector<std::array<double, 3> > serif::geometry::deformation::ReferenceSurface::positions

10.420.2.2 scalar_dofs

std::vector<int> serif::geometry::deformation::ReferenceSurface::scalar_dofs

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/reference_surface.hpp

10.421 serif::operators::backend::mfem::volume_keys::ReferenceValues< Contribution, OperandT > Struct Template Reference

#include <volume_data.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.422 serif::eos::relations::Relation< Output, Inputs > Struct Template Reference

#include <catalog.hpp>

Public Types

- using OutputQuantity = Output
- using InputQuantities = dimensions::QuantityList<Inputs...>

Static Public Attributes

- static constexpr std::size_t inputCount = CountVariadicArguments<Inputs...>()

10.422.1 Member Typedef Documentation

10.422.1.1 InputQuantities

template<typename Output, typename... Inputs>
using serif::eos::relations::Relation< Output, Inputs >::InputQuantities = dimensions::QuantityList<Inputs...>

10.422.1.2 OutputQuantity

template<typename Output, typename... Inputs>
using serif::eos::relations::Relation< Output, Inputs >::OutputQuantity = Output

10.422.2 Member Data Documentation

10.422.2.1 inputCount

template<typename Output, typename... Inputs>
std::size_t serif::eos::relations::Relation< Output, Inputs >::inputCount = CountVariadicArguments<Inputs...>() [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/catalog.hpp

10.423 serif::eos::relations::RelationCatalog< Relations > Struct Template Reference

#include <catalog.hpp>

Static Public Attributes

- static constexpr std::size_t size = CountVariadicArguments<Relations...>()

10.423.1 Member Data Documentation

10.423.1.1 size

template<typename... Relations>
std::size_t serif::eos::relations::RelationCatalog< Relations >::size = CountVariadicArguments<Relations...>() [static], [constexpr]

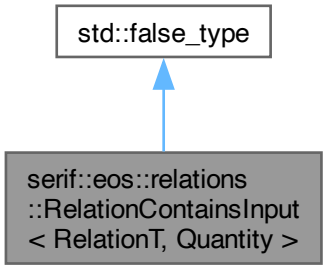
The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/catalog.hpp

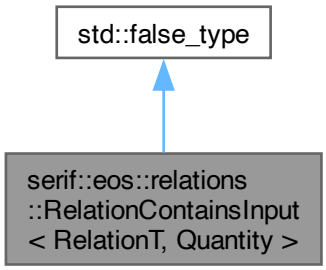
10.424 serif::eos::relations::RelationContainsInput< RelationT, Quantity > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::relations::RelationContainsInput< RelationT, Quantity >:



Collaboration diagram for serif::eos::relations::RelationContainsInput< RelationT, Quantity >:



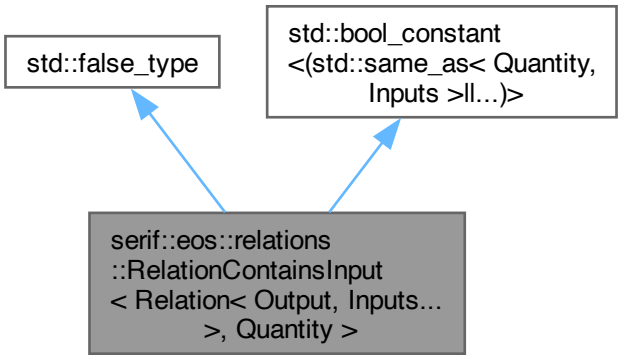
The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/traits.hpp

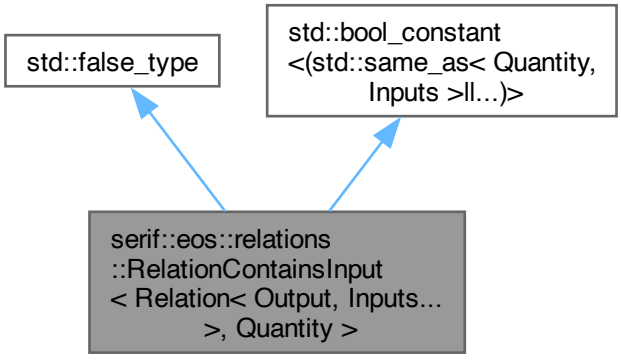
10.425 serif::eos::relations::RelationContainsInput< Relation< Output, Inputs... >, Quantity > Struct Template Reference

#include <traits.hpp>

Inheritance diagram for serif::eos::relations::RelationContainsInput< Relation< Output, Inputs... >, Quantity >:



Collaboration diagram for serif::eos::relations::RelationContainsInput< Relation< Output, Inputs... >, Quantity >:



The documentation for this struct was generated from the following file:

- src/include/serif/eos/relations/traits.hpp

10.426 serif::discretization::domain::relation::lists::RelationList< RelationTs > Struct Template Reference

#include <relation_list.hpp>

Static Public Attributes

- static constexpr std::size_t count = CountVariadicArguments<RelationTs...>()

10.426.1 Member Data Documentation

10.426.1.1 count

template<IsRelation... RelationTs>
std::size_t serif::discretization::domain::relation::lists::RelationList< RelationTs >::count = CountVariadicArguments<RelationTs...>() [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/lists/relation_list.hpp

10.427 serif::discretization::domain::schema::RelationList< RelationTs > Struct Template Reference

#include <relation_list.hpp>

Static Public Attributes

- static constexpr std::size_t count = CountVariadicArguments<RelationTs...>()

10.427.1 Member Data Documentation

10.427.1.1 count

template<IsRelation... RelationTs>
std::size_t serif::discretization::domain::relation::lists::RelationList< RelationTs >::count = CountVariadicArguments<RelationTs...>() [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/lists/relation_list.hpp

10.428 serif::geometry::surface::detail::RelationQuantities< Relation > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/detail/relations.hpp

10.429 serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > > Struct Template Reference

#include <relations.hpp>

Public Types

- using AllInputs = utils::dependencies::KeyList<Inputs...>
- using NonPressureInputs

10.429.1 Member Typedef Documentation

10.429.1.1 AllInputs

template<typename Output, typename... Inputs>
using serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > >::AllInputs = utils::dependencies::KeyList<Inputs...>

10.429.1.2 NonPressureInputs

template<typename Output, typename... Inputs>
using serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > >::NonPressureInputs

Initial value:

```
utils::dependencies::CombinedKeyList<
        std::conditional_t<std::same_as<Inputs, dimensions::Pressure>, utils::dependencies::KeyList<>,
    utils::dependencies::KeyList<Inputs>>...>
```

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/detail/relations.hpp

10.430 serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, RelationListT > Struct Template Reference

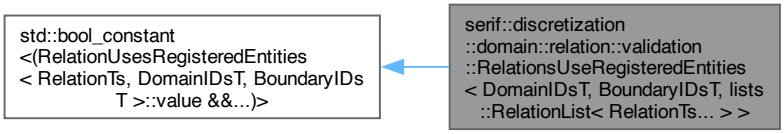
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/validation.hpp

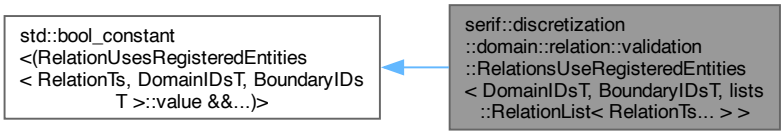
10.431 serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, lists::RelationList< RelationTs... > > Struct Template Reference

#include <validation.hpp>

Inheritance diagram for serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, lists::RelationList< RelationTs... > >:



Collaboration diagram for serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT, lists::RelationList< RelationTs... > >:



The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/validation.hpp

10.432 serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< RelationT, DomainIDsT, BoundaryIDsT > Struct Template Reference

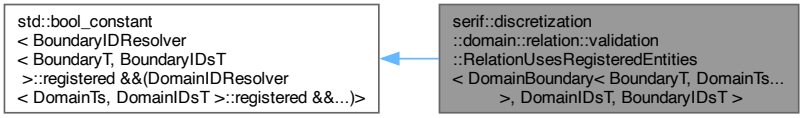
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/validation.hpp

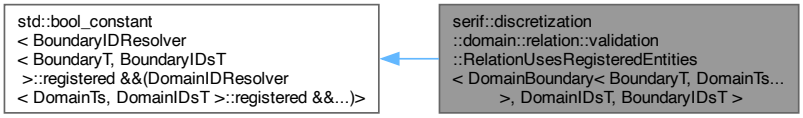
10.433 serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< DomainBoundary< BoundaryT, DomainTs... >, DomainIDsT, BoundaryIDsT > Struct Template Reference

#include <validation.hpp>

Inheritance diagram for serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< DomainBoundary< BoundaryT, DomainTs... >, DomainIDsT, BoundaryIDsT >:



Collaboration diagram for serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< DomainBoundary< BoundaryT, DomainTs... >, DomainIDsT, BoundaryIDsT >:



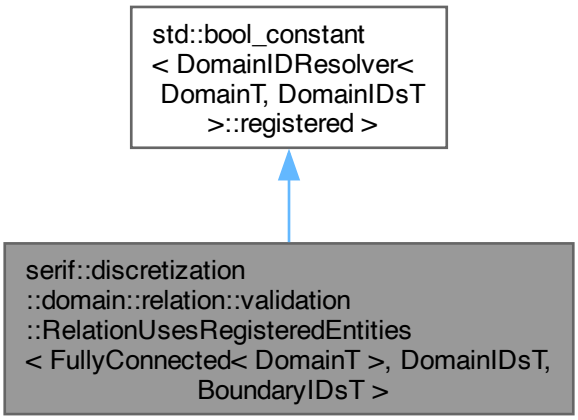
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/validation.hpp

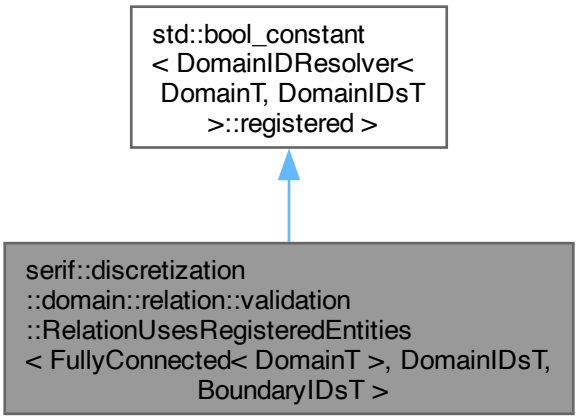
10.434 serif::discretization::domain::relation::validation::RelationUses↩ RegisteredEntities< FullyConnected< DomainT >, DomainIDsT, BoundaryIDsT > Struct Template Reference

#include <validation.hpp>

Inheritance diagram for serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< FullyConnected< DomainT >, DomainIDsT, BoundaryIDsT >:



Collaboration diagram for serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< FullyConnected< DomainT >, DomainIDsT, BoundaryIDsT >:



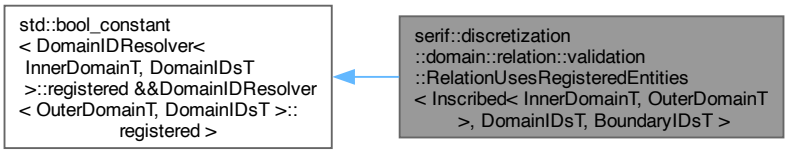
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/validation.hpp

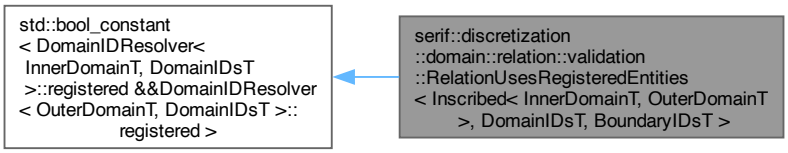
10.435 serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< Inscribed< InnerDomainT, OuterDomainT >, DomainIDsT, BoundaryIDsT > Struct Template Reference

#include <validation.hpp>

Inheritance diagram for serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< Inscribed< InnerDomainT, OuterDomainT >, DomainIDsT, BoundaryIDsT >:



Collaboration diagram for serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< Inscribed< InnerDomainT, OuterDomainT >, DomainIDsT, BoundaryIDsT >:



The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/validation.hpp

10.436 serif::discretization::domain::relation::validation::RelationValidationResult Struct Reference

#include <runtime.hpp>

Classes

- struct ConnectedDiagnostics
- struct DomainBoundaryDiagnostics
- struct InscribedDiagnostics

Public Member Functions

- bool valid () const noexcept
- operator bool () const noexcept

Public Attributes

- RelationValidationFailure failure {RelationValidationFailure::None}
- std::optional< InscribedDiagnostics > inscribedDiagnostics = std::nullopt
- std::optional< ConnectedDiagnostics > connectedDiagnostics = std::nullopt
- std::optional< DomainBoundaryDiagnostics > domainBoundaryDiagnostics = std::nullopt

10.436.1 Member Function Documentation

10.436.1.1 operator bool()

serif::discretization::domain::relation::validation::RelationValidationResult::operator bool () const [inline], [explicit], [nodiscard], [noexcept]

10.436.1.2 valid()

bool serif::discretization::domain::relation::validation::RelationValidationResult::valid () const [inline], [nodiscard], [noexcept]

10.436.2 Member Data Documentation

10.436.2.1 connectedDiagnostics

std::optional<ConnectedDiagnostics> serif::discretization::domain::relation::validation::RelationValidationResult::connected↩Diagnostics = std::nullopt

10.436.2.2 domainBoundaryDiagnostics

std::optional<DomainBoundaryDiagnostics> serif::discretization::domain::relation::validation::RelationValidationResult↩::domainBoundaryDiagnostics = std::nullopt

10.436.2.3 failure

RelationValidationFailure serif::discretization::domain::relation::validation::RelationValidationResult::failure {RelationValidationFailure::None}

10.436.2.4 inscribedDiagnostics

std::optional<InscribedDiagnostics> serif::discretization::domain::relation::validation::RelationValidationResult::inscribed↩
Diagnostics = std::nullopt

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/runtime.hpp

10.437 serif::discretization::domain::schema::validation::Relation↩ ValidationResult Struct Reference

#include <runtime.hpp>

Classes

- struct ConnectedDiagnostics
- struct DomainBoundaryDiagnostics
- struct InscribedDiagnostics

Public Member Functions

- bool valid () const noexcept
- operator bool () const noexcept

Public Attributes

- RelationValidationFailure failure {RelationValidationFailure::None}
- std::optional< InscribedDiagnostics > inscribedDiagnostics = std::nullopt
- std::optional< ConnectedDiagnostics > connectedDiagnostics = std::nullopt
- std::optional< DomainBoundaryDiagnostics > domainBoundaryDiagnostics = std::nullopt

10.437.1 Member Function Documentation

10.437.1.1 operator bool()

serif::discretization::domain::relation::validation::RelationValidationResult::operator bool () const [inline], [explicit], [nodiscard], [noexcept]

10.437.1.2 valid()

bool serif::discretization::domain::relation::validation::RelationValidationResult::valid () const [inline], [nodiscard], [noexcept]

10.437.2 Member Data Documentation

10.437.2.1 connectedDiagnostics

std::optional<ConnectedDiagnostics> serif::discretization::domain::relation::validation::RelationValidationResult::connected↩
Diagnostics = std::nullopt

10.437.2.2 domainBoundaryDiagnostics

std::optional<DomainBoundaryDiagnostics> serif::discretization::domain::relation::validation::RelationValidationResult↩
::domainBoundaryDiagnostics = std::nullopt

10.437.2.3 failure

RelationValidationFailure serif::discretization::domain::relation::validation::RelationValidationResult::failure {RelationValidationFailure::None}

10.437.2.4 inscribedDiagnostics

std::optional<InscribedDiagnostics> serif::discretization::domain::relation::validation::RelationValidationResult::inscribed↩
Diagnostics = std::nullopt

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/relation/validation/runtime.hpp

10.438 serif::discretization::domain::schema::validation::RelationValidator< RelationT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/schema/validation/validator.hpp

10.439 serif::discretization::domain::schema::validation::RelationValidator< DomainBoundary< BoundaryT, DomainTs... > > Struct Template Reference

#include <boundary.hpp>

Static Public Member Functions

- template<IsSchema SchemaT>
  static RelationValidationResult validate (const mesh::MeshTopology &topology)

10.439.1 Member Function Documentation

10.439.1.1 validate()

template<IsBoundary BoundaryT, IsDomainOrSet... DomainTs>
template<IsSchema SchemaT>
static RelationValidationResult serif::discretization::domain::schema::validation::RelationValidator< DomainBoundary< BoundaryT, DomainTs... > >::validate (
const mesh::MeshTopology & topology) [inline], [static], [nodiscard]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/schema/validation/boundary.hpp

10.440 serif::discretization::domain::schema::validation::RelationValidator< FullyConnected< DomainT > > Struct Template Reference

#include <connected.hpp>

Static Public Member Functions

- template<IsSchema SchemaT>
  static RelationValidationResult validate (const mesh::MeshTopology &topology)

10.440.1 Member Function Documentation

10.440.1.1 validate()

template<IsDomainOrSet DomainT>
template<IsSchema SchemaT>
static RelationValidationResult serif::discretization::domain::schema::validation::RelationValidator< FullyConnected< DomainT > >::validate (
const mesh::MeshTopology & topology) [inline], [static], [nodiscard]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/schema/validation/connected.hpp

10.441 serif::discretization::domain::schema::validation::RelationValidator< Inscribed< InnerT, OuterT > > Struct Template Reference

#include <inscribed.hpp>

Static Public Member Functions

- template<IsSchema SchemaT>
  static RelationValidationResult validate (const mesh::MeshTopology &topology)

10.441.1 Member Function Documentation

10.441.1.1 validate()

template<IsDomainOrSet InnerT, IsDomainOrSet OuterT>
template<IsSchema SchemaT>
static RelationValidationResult serif::discretization::domain::schema::validation::RelationValidator< Inscribed< InnerT, OuterT > >::validate (
const mesh::MeshTopology & topology) [inline], [static], [nodiscard]

The documentation for this struct was generated from the following file:

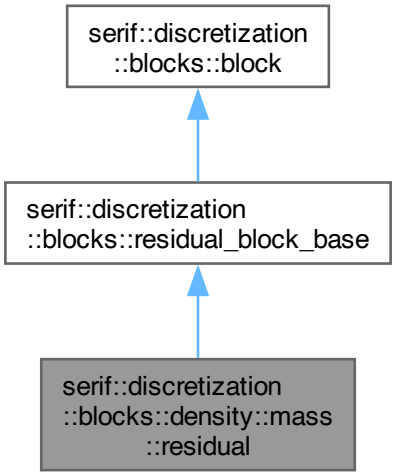
- src/include/serif/discretization/domain/schema/validation/inscribed.hpp

10.442 serif::discretization::blocks::density::mass::residual Struct Reference

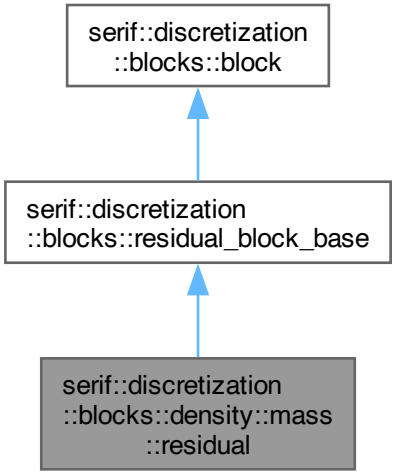
The residual block is a block that contains the residual of the mass density term. It is a scalar field that is associated with the mass density of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::density::mass::residual:



Collaboration diagram for serif::discretization::blocks::density::mass::residual:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::residual_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.442.1 Detailed Description

The residual block is a block that contains the residual of the mass density term. It is a scalar field that is associated with the mass density of the system.

The documentation for this struct was generated from the following file:

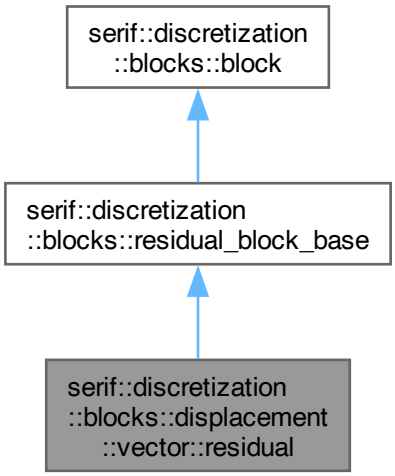
- src/include/serif/discretization/blocks/fields.hpp

10.443 serif::discretization::blocks::displacement::vector::residual Struct Reference

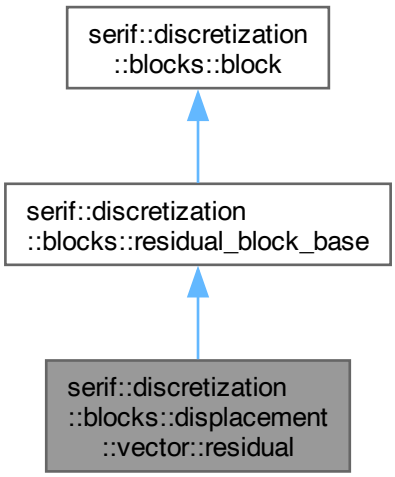
The residual block is a block that contains the residual of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::displacement::vector::residual:



Collaboration diagram for serif::discretization::blocks::displacement::vector::residual:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::residual_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.443.1 Detailed Description

The residual block is a block that contains the residual of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces.

The documentation for this struct was generated from the following file:

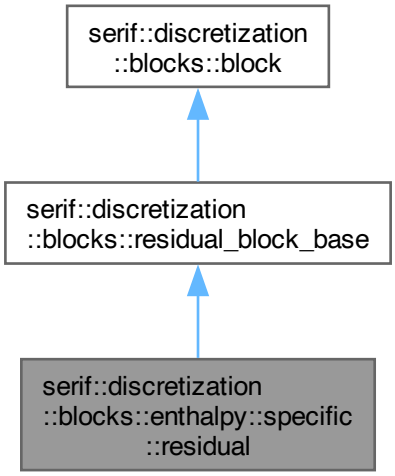
- src/include/serif/discretization/blocks/fields.hpp

10.444 serif::discretization::blocks::enthalpy::specific::residual Struct Reference

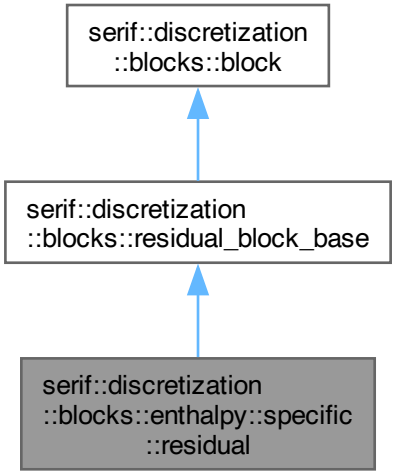
The residual block is a block that contains the residual of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::enthalpy::specific::residual:



Collaboration diagram for serif::discretization::blocks::enthalpy::specific::residual:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::residual_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.444.1 Detailed Description

The residual block is a block that contains the residual of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system.

The documentation for this struct was generated from the following file:

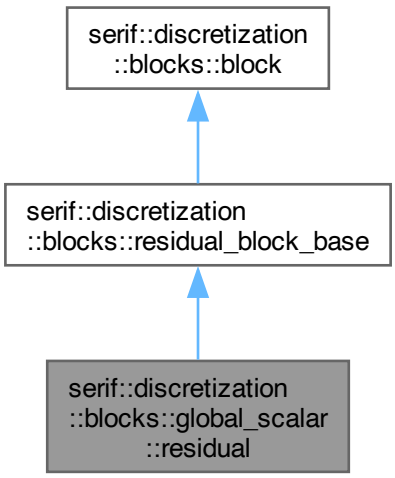
- src/include/serif/discretization/blocks/fields.hpp

10.445 serif::discretization::blocks::global_scalar::residual Struct Reference

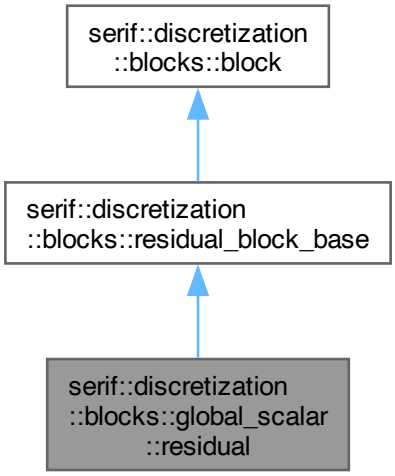
The residual block is a block that contains the residual of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the residual of the mass of the system (i.e. the difference between the current mass and the target mass).

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::global_scalar::residual:



Collaboration diagram for serif::discretization::blocks::global_scalar::residual:



Static Public Attributes

- static constexpr int static_block_size = 1

  Because this is a global scalar unknown we can be sure at compile time that the block size is 1 and we can use this to optimize the linear algebra operations on this block.

Static Public Attributes inherited from serif::discretization::blocks::residual_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.445.1 Detailed Description

The residual block is a block that contains the residual of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the residual of the mass of the system (i.e. the difference between the current mass and the target mass).

10.445.2 Member Data Documentation

10.445.2.1 static_block_size

int serif::discretization::blocks::global_scalar::residual::static_block_size = 1 [static], [constexpr]

Because this is a global scalar unknown we can be sure at compile time that the block size is 1 and we can use this to optimize the linear algebra operations on this block.

The documentation for this struct was generated from the following file:

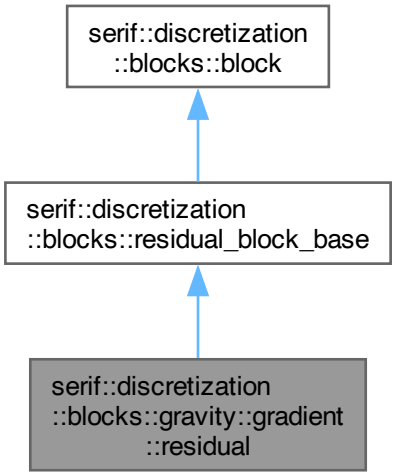
- src/include/serif/discretization/blocks/fields.hpp

10.446 serif::discretization::blocks::gravity::gradient::residual Struct Reference

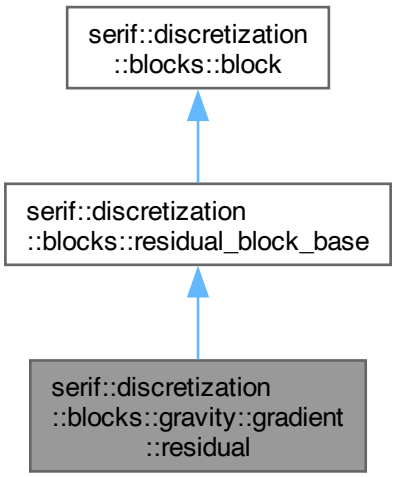
The residual block is a block that contains the residual of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::gravity::gradient::residual:



Collaboration diagram for serif::discretization::blocks::gravity::gradient::residual:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::residual_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.446.1 Detailed Description

The residual block is a block that contains the residual of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system.

The documentation for this struct was generated from the following file:

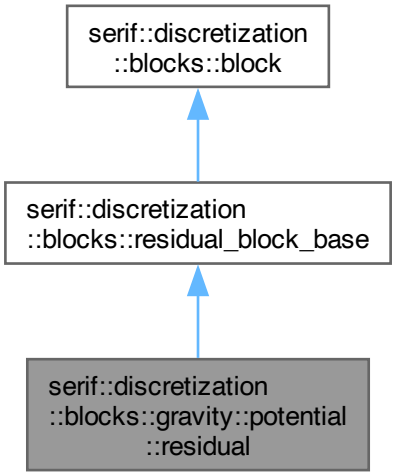
- src/include/serif/discretization/blocks/fields.hpp

10.447 serif::discretization::blocks::gravity::potential::residual Struct Reference

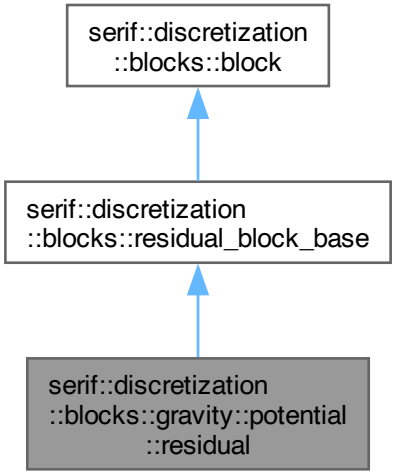
The residual block is a block that contains the residual of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::gravity::potential::residual:



Collaboration diagram for serif::discretization::blocks::gravity::potential::residual:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::residual_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.447.1 Detailed Description

The residual block is a block that contains the residual of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system.

The documentation for this struct was generated from the following file:

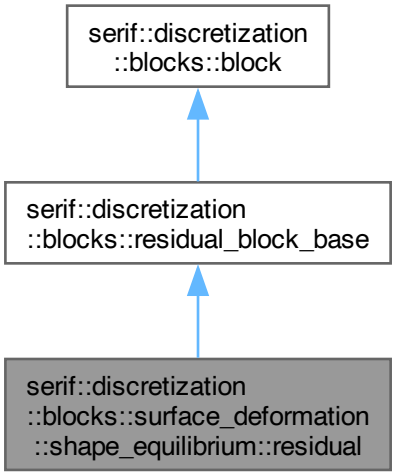
- src/include/serif/discretization/blocks/fields.hpp

10.448 serif::discretization::blocks::surface_deformation::shape_equilibrium::residual Struct Reference

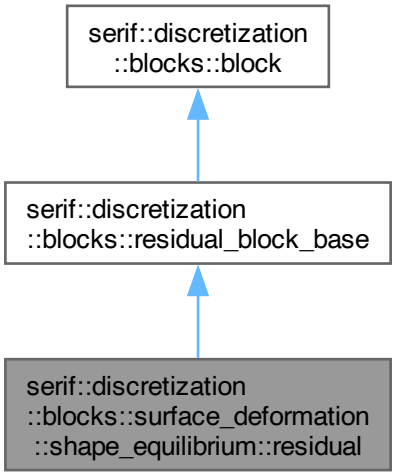
The residuals of the shape equilibrium condition.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::surface_deformation::shape_equilibrium::residual:



Collaboration diagram for serif::discretization::blocks::surface_deformation::shape_equilibrium::residual:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::residual_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.448.1 Detailed Description

The residuals of the shape equilibrium condition.

The documentation for this struct was generated from the following file:

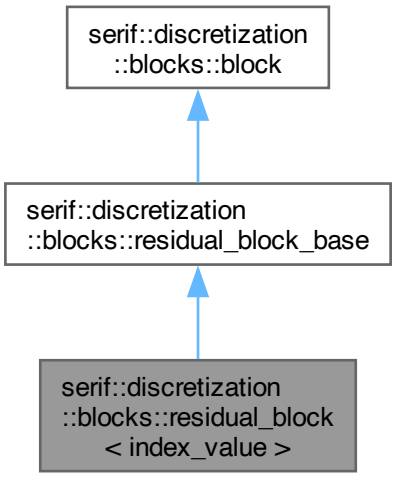
- src/include/serif/discretization/blocks/fields.hpp

10.449 serif::discretization::blocks::residual_block< index_value > Struct Template Reference

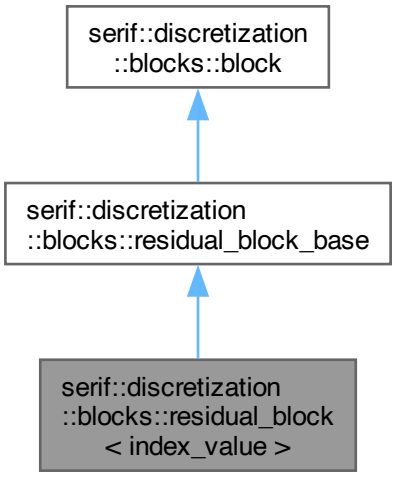
Final class for residual blocks with a fixed index, derived from residual_block_base.

#include <base.hpp>

Inheritance diagram for serif::discretization::blocks::residual_block< index_value >:



Collaboration diagram for serif::discretization::blocks::residual_block< index_value >:



Public Member Functions

- constexpr operator int () const noexcept

Static Public Attributes

- static constexpr int index = index_value

Static Public Attributes inherited from serif::discretization::blocks::residual_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.449.1 Detailed Description

template<int index_value>
struct serif::discretization::blocks::residual_block< index_value >

Final class for residual blocks with a fixed index, derived from residual_block_base.

10.449.2 Member Function Documentation

10.449.2.1 operator int()

template<int index_value>
serif::discretization::blocks::residual_block< index_value >::operator int () const [inline], [constexpr], [noexcept]

10.449.3 Member Data Documentation

10.449.3.1 index

template<int index_value>
int serif::discretization::blocks::residual_block< index_value >::index = index_value [static], [constexpr]

The documentation for this struct was generated from the following file:

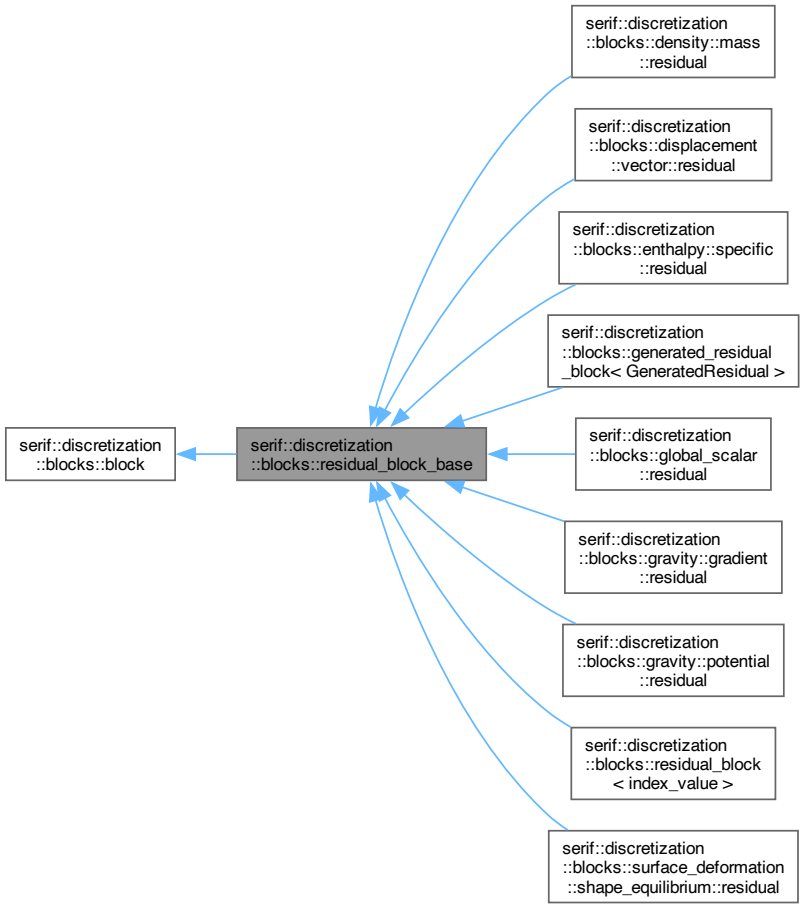
- src/include/serif/discretization/blocks/base.hpp

10.450 serif::discretization::blocks::residual_block_base Struct Reference

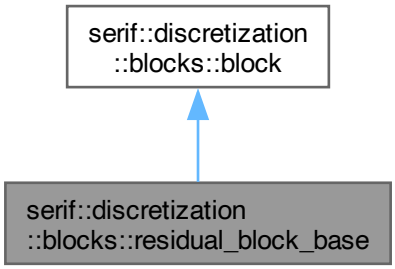
Base class for residual blocks, derived from block.

#include <base.hpp>

Inheritance diagram for serif::discretization::blocks::residual_block_base:



Collaboration diagram for serif::discretization::blocks::residual_block_base:



Static Public Attributes

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.450.1 Detailed Description

Base class for residual blocks, derived from block.

10.450.2 Member Data Documentation

10.450.2.1 static_block_size

int serif::discretization::blocks::residual_block_base::static_block_size = DYNAMIC_BLOCK_SIZE [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/base.hpp

10.451 serif::discretization::quadrature::Resolution Struct Reference

#include <rules.hpp>

Public Attributes

- int base_order
- int boost
- int order
- bool used_fixed_order

10.451.1 Member Data Documentation

10.451.1.1 base_order

int serif::discretization::quadrature::Resolution::base_order

10.451.1.2 boost

int serif::discretization::quadrature::Resolution::boost

10.451.1.3 order

int serif::discretization::quadrature::Resolution::order

10.451.1.4 used_fixed_order

bool serif::discretization::quadrature::Resolution::used_fixed_order

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

10.452 serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, Available > Struct Template Reference

#include <schema.hpp>

Static Public Attributes

- static constexpr bool valid = false

10.452.1 Member Data Documentation

10.452.1.1 valid

template<typename ComputationT, typename BindingsT, bool Available = SatisfiesKeyList<BindingsT, typename ComputationT::Inputs>>
bool serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, Available >::valid = false [static], [constexpr]

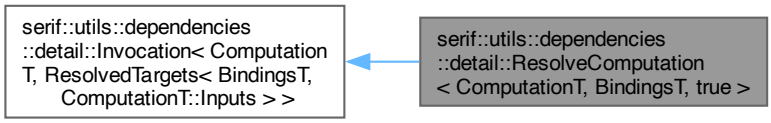
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

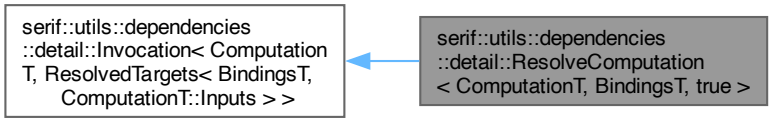
10.453 serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, true > Struct Template Reference

#include <schema.hpp>

Inheritance diagram for serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, true >:



Collaboration diagram for serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, true >:



Static Public Attributes

- static constexpr bool valid

Static Public Attributes inherited from serif::utils::dependencies::detail::Invocation< ComputationT, ResolvedTargets< BindingsT, ComputationT

- static constexpr bool valid

10.453.1 Member Data Documentation

10.453.1.1 valid

bool serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, Available >::valid [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

10.454 serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, Done > Struct Template Reference

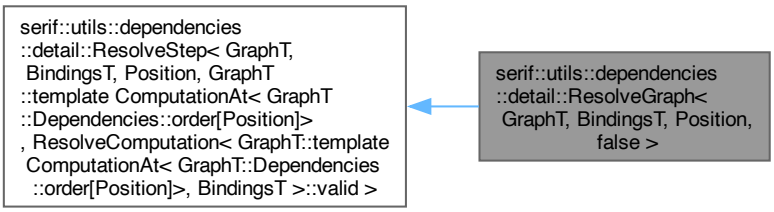
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

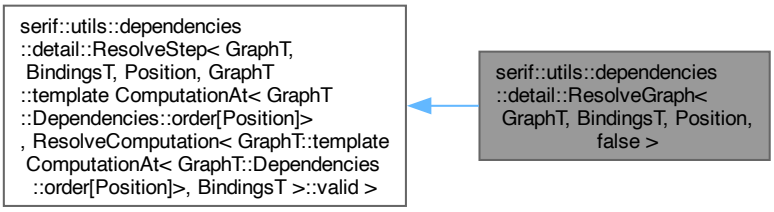
10.455 serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, false > Struct Template Reference

#include <schema.hpp>

Inheritance diagram for serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, false >:



Collaboration diagram for serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, false >:



Additional Inherited Members

Static Public Attributes inherited from serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, GraphT::template Computa

- static constexpr bool valid

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

10.456 serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true > Struct Template Reference

#include <schema.hpp>

Public Types

- using ResolvedBindings = BindingsT

Static Public Attributes

- static constexpr bool valid = true

10.456.1 Member Typedef Documentation

10.456.1.1 ResolvedBindings

template<typename GraphT, typename BindingsT, std::size_t Position>
using serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true >::ResolvedBindings = BindingsT

10.456.2 Member Data Documentation

10.456.2.1 valid

template<typename GraphT, typename BindingsT, std::size_t Position>
bool serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true >::valid = true [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

10.457 serif::utils::dependencies::ResolveKeys< BindingsT, KeysT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/key_list.hpp

10.458 serif::utils::dependencies::ResolveKeys< BindingsT, KeyList< KeyTs... > > Struct Template Reference

#include <key_list.hpp>

Public Types

- using type = types::TypeList<bound_type_t<BindingsT, KeyTs>...>

10.458.1 Member Typedef Documentation

10.458.1.1 type

template<typename BindingsT, typename... KeyTs>
using serif::utils::dependencies::ResolveKeys< BindingsT, KeyList< KeyTs... > >::type = types::TypeList<bound_type_t<BindingsT, KeyTs>...>

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/key_list.hpp

10.459 serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, Valid > Struct Template Reference

#include <schema.hpp>

Static Public Attributes

- static constexpr bool valid = false

10.459.1 Member Data Documentation

10.459.1.1 valid

template<typename GraphT, typename BindingsT, std::size_t Position, typename ComputationT, bool Valid>
bool serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, Valid >::valid = false [static], [constexpr]

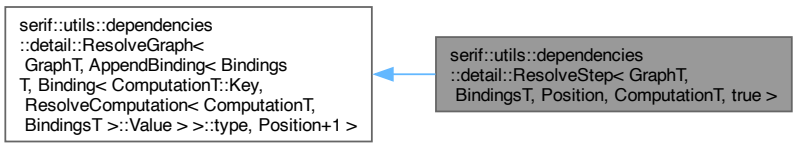
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

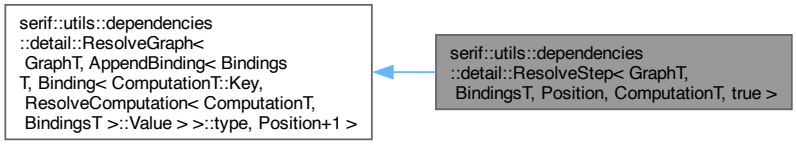
10.460 serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, true > Struct Template Reference

#include <schema.hpp>

Inheritance diagram for serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, true >:



Collaboration diagram for serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, true >:



Static Public Attributes

- static constexpr bool valid

10.460.1 Member Data Documentation

10.460.1.1 valid

bool serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, Valid >::valid [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/schema.hpp

10.461 serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT > Class Template Reference

#include <resource.hpp>

Public Types

- using Key = KeyT
- using Target = typename StorageT::Target
- using Policy = PolicyT

Public Member Functions

- Resource (StorageT storage, PolicyT policy)

  Keep storage and its tracking policy together under one role key.
- const Target & get () const

  Read the resource without allowing computations to mutate their inputs.
- const PolicyT & policy () const

  Expose tracking separately from the value passed to a computation.

Private Attributes

- StorageT m_storage
- PolicyT m_policy

10.461.1 Member Typedef Documentation

10.461.1.1 Key

template<typename KeyT, typename StorageT, typename PolicyT>
using serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >::Key = KeyT

10.461.1.2 Policy

template<typename KeyT, typename StorageT, typename PolicyT>
using serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >::Policy = PolicyT

10.461.1.3 Target

template<typename KeyT, typename StorageT, typename PolicyT>
using serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >::Target = typename StorageT::Target

10.461.2 Constructor & Destructor Documentation

10.461.2.1 Resource()

template<typename KeyT, typename StorageT, typename PolicyT>
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >::Resource (
StorageT storage,
PolicyT policy) [inline]

Keep storage and its tracking policy together under one role key.

10.461.3 Member Function Documentation

10.461.3.1 get()

template<typename KeyT, typename StorageT, typename PolicyT>
const Target & serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >::get () const [inline], [nodiscard]

Read the resource without allowing computations to mutate their inputs.

10.461.3.2 policy()

template<typename KeyT, typename StorageT, typename PolicyT>
const PolicyT & serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >::policy () const [inline], [nodiscard]

Expose tracking separately from the value passed to a computation.

10.461.4 Member Data Documentation

10.461.4.1 m_policy

template<typename KeyT, typename StorageT, typename PolicyT>
PolicyT serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >::m_policy [private]

10.461.4.2 m_storage

template<typename KeyT, typename StorageT, typename PolicyT>
StorageT serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >::m_storage [private]

The documentation for this class was generated from the following file:

- src/include/serif/utils/dependencies/resource.hpp

10.462 serif::utils::dependencies::detail::ResourceKeys< ResourcesT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_traits.hpp

10.463 serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > > Struct Template Reference

#include <cache_traits.hpp>

Public Types

- using type = KeyList<typename Resources::Key...>

Static Public Attributes

- static constexpr bool trackable = (SnapshotPolicyFor<typename Resources::Policy, typename Resources::Target> && ...)

10.463.1 Member Typedef Documentation

10.463.1.1 type

template<typename... Resources>
using serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > >::type = KeyList<typename Resources::Key...>

10.463.2 Member Data Documentation

10.463.2.1 trackable

template<typename... Resources>
bool serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > >::trackable = (SnapshotPolicy↩
For<typename Resources::Policy, typename Resources::Target> && ...) [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_traits.hpp

10.464 serif::utils::dependencies::detail::Result< T > Struct Template Reference

#include <result.hpp>

Public Types

- using Value = T
- using Error = void

Static Public Attributes

- static constexpr bool fallible = false
- static constexpr bool valid = std::is_object_v<T> && std::move_constructible<T>

10.464.1 Member Typedef Documentation

10.464.1.1 Error

template<typename T>
using serif::utils::dependencies::detail::Result< T >::Error = void

10.464.1.2 Value

template<typename T>
using serif::utils::dependencies::detail::Result< T >::Value = T

10.464.2 Member Data Documentation

10.464.2.1 fallible

template<typename T>
bool serif::utils::dependencies::detail::Result< T >::fallible = false [static], [constexpr]

10.464.2.2 valid

template<typename T>
bool serif::utils::dependencies::detail::Result< T >::valid = std::is_object_v<T> && std::move_constructible<T> [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/result.hpp

10.465 serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > > Struct Template Reference

#include <result.hpp>

Public Types

- using Value = ValueT
- using Error = ErrorT
- using Value
- using Error

Static Public Attributes

- static constexpr bool fallible = true
- static constexpr bool valid = std::is_object_v<ValueT> && std::move_constructible<ValueT>
- static constexpr bool fallible
- static constexpr bool valid

10.465.1 Member Typedef Documentation

10.465.1.1 Error [1/2]

using serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >::Error

10.465.1.2 Error [2/2]

template<typename ValueT, typename ErrorT>
using serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >::Error = ErrorT

10.465.1.3 Value [1/2]

using serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >::Value

10.465.1.4 Value [2/2]

template<typename ValueT, typename ErrorT>
using serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >::Value = ValueT

10.465.2 Member Data Documentation

10.465.2.1 fallible [1/2]

bool serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >::fallible [static], [constexpr]

10.465.2.2 fallible [2/2]

template<typename ValueT, typename ErrorT>
bool serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >::fallible = true [static], [constexpr]

10.465.2.3 valid [1/2]

bool serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >::valid [static], [constexpr]

10.465.2.4 valid [2/2]

template<typename ValueT, typename ErrorT>
bool serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >::valid = std::is_object_v<ValueT> && std::move_constructible<ValueT> [static], [constexpr]

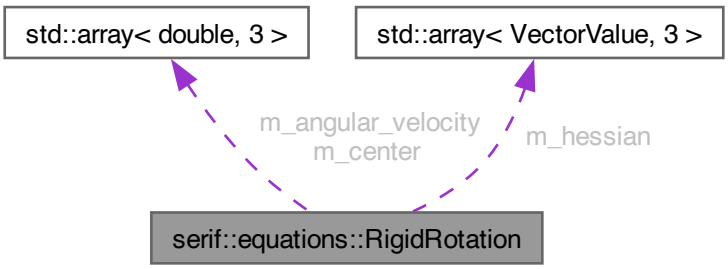
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/result.hpp

10.466 serif::equations::RigidRotation Class Reference

#include <rigid_rotation.hpp>

Collaboration diagram for serif::equations::RigidRotation:



Public Types

- using Vector = discretization::forms::VectorValue
- using Matrix = discretization::forms::MatrixValue

Public Member Functions

- RigidRotation (Vector angular_velocity, Vector center={})

  Define an arbitrary rotation axis through center; reject nonfinite parameters or squared speed.
- double potential (const Vector &position) const

  Evaluate the nonnegative potential without subtracting nearly equal squared lengths, this is an important thing to do otherwise we get round off noise.
- Vector gradient (const Vector &position) const

  grad(Psi) = H (x - center), with H = |omega|^2 I - omega tensor omega.
- const Matrix & hessian () const

  The position derivative of grad(Psi) is constant for rigid rotation.
- const Vector & angular_velocity () const

  Physical angular-velocity vector represented by this value utility.
- const Vector & center () const

  A point on the rotation axis, in physical coordinates.

Static Public Member Functions

- static RigidRotation unit_rotation (Vector axis={0.0, 0.0, 1.0}, Vector center={})

  Unit-speed rotation about a prescribed nonzero axis and center.

Private Attributes

- Vector m_angular_velocity
- Vector m_center
- Matrix m_hessian {}

10.466.1 Detailed Description

Positive centrifugal potential Psi = |omega cross (x - center)|^2 / 2. Its gradient is the outward acceleration; displacement balance uses -rho grad(Psi). A value utility; equations bind their angular speed separately as an unknown.

10.466.2 Member Typedef Documentation

10.466.2.1 Matrix

using serif::equations::RigidRotation::Matrix = discretization::forms::MatrixValue

10.466.2.2 Vector

using serif::equations::RigidRotation::Vector = discretization::forms::VectorValue

10.466.3 Constructor & Destructor Documentation

10.466.3.1 RigidRotation()

serif::equations::RigidRotation::RigidRotation (
Vector angular_velocity,
Vector center = {}) [inline], [explicit]

Define an arbitrary rotation axis through center; reject nonfinite parameters or squared speed.

10.466.4 Member Function Documentation

10.466.4.1 angular_velocity()

const Vector & serif::equations::RigidRotation::angular_velocity () const [inline], [nodiscard]

Physical angular-velocity vector represented by this value utility.

10.466.4.2 center()

const Vector & serif::equations::RigidRotation::center () const [inline], [nodiscard]

A point on the rotation axis, in physical coordinates.

10.466.4.3 gradient()

Vector serif::equations::RigidRotation::gradient (
const Vector & position) const [inline], [nodiscard]

grad(Psi) = H (x - center), with H = |omega|^2 I - omega tensor omega.

10.466.4.4 hessian()

const Matrix & serif::equations::RigidRotation::hessian () const [inline], [nodiscard]

The position derivative of grad(Psi) is constant for rigid rotation.

10.466.4.5 potential()

double serif::equations::RigidRotation::potential (
const Vector & position) const [inline], [nodiscard]

Evaluate the nonnegative potential without subtracting nearly equal squared lengths, this is an important thing to do otherwise we get round off noise.

10.466.4.6 unit_rotation()

static RigidRotation serif::equations::RigidRotation::unit_rotation (
Vector axis = {0.0, 0.0, 1.0},
Vector center = {}) [inline], [static], [nodiscard]

Unit-speed rotation about a prescribed nonzero axis and center.

10.466.5 Member Data Documentation

10.466.5.1 m_angular_velocity

Vector serif::equations::RigidRotation::m_angular_velocity [private]

10.466.5.2 m_center

Vector serif::equations::RigidRotation::m_center [private]

10.466.5.3 m_hessian

Matrix serif::equations::RigidRotation::m_hessian {} [private]

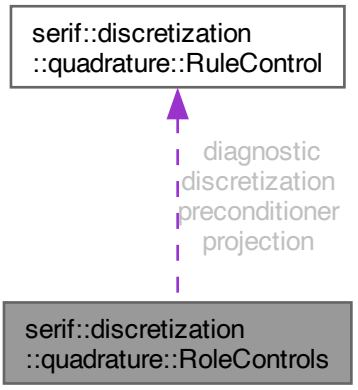
The documentation for this class was generated from the following file:

- src/include/serif/equations/rigid_rotation.hpp

10.467 serif::discretization::quadrature::RoleControls Struct Reference

#include <rules.hpp>

Collaboration diagram for serif::discretization::quadrature::RoleControls:



Public Attributes

- RuleControl discretization
- RuleControl preconditioner
- RuleControl diagnostic
- RuleControl projection

10.467.1 Member Data Documentation

10.467.1.1 diagnostic

RuleControl serif::discretization::quadrature::RoleControls::diagnostic

10.467.1.2 discretization

RuleControl serif::discretization::quadrature::RoleControls::discretization

10.467.1.3 preconditioner

RuleControl serif::discretization::quadrature::RoleControls::preconditioner

10.467.1.4 projection

RuleControl serif::discretization::quadrature::RoleControls::projection

The documentation for this struct was generated from the following file:

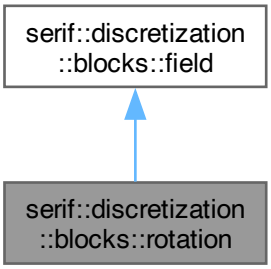
- src/include/serif/discretization/quadrature/rules.hpp

10.468 serif::discretization::blocks::rotation Struct Reference

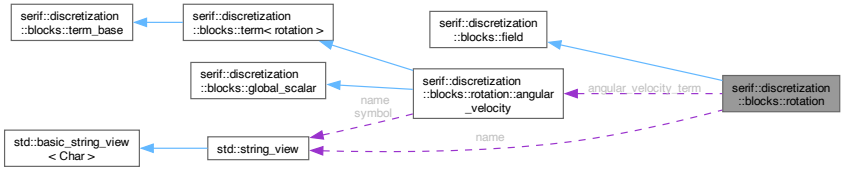
The rotational field, represented as the angular velocity term, given that we currently only model rigid rotation this is a global scalar. A future extension which models differential rotation will likely need to modify this to some vector field or other more complex representation of the rotation.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::rotation:



Collaboration diagram for serif::discretization::blocks::rotation:



Classes

- struct angular_velocity

  The angular_velocity term represents the angular velocity of the system. It is a global scalar that is associated with the rotation of the system. This is used to enforce the rotation condition in the weak form when we are in the rigid rotation case. Note that downstream the way that this currently works is to calculate the expected angular velocity given the current mass, geometry, density, and angular momentum and then prescribe, given those, what the angular velocity should be. Therefore, this does not enforce that the angular velocity must be constant over the entire solver run, but rather that the angular velocity is consistent with the current state of the system.

Public Types

- using Terms = utils::types::TypeList<angular_velocity>

  Type list of terms for compile time reflection and iteration over the terms of the field.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<rotation>()

  Compile time reflected name of the field.
- static constexpr angular_velocity angular_velocity_term {}

  Static constexpr instance of the angular_velocity term. This is used to access the term in the discretization.

10.468.1 Detailed Description

The rotational field, represented as the angular velocity term, given that we currently only model rigid rotation this is a global scalar. A future extension which models differential rotation will likely need to modify this to some vector field or other more complex representation of the rotation.

10.468.2 Member Typedef Documentation

10.468.2.1 Terms

using serif::discretization::blocks::rotation::Terms = utils::types::TypeList<angular_velocity>

Type list of terms for compile time reflection and iteration over the terms of the field.

10.468.3 Member Data Documentation

10.468.3.1 angular_velocity_term

angular_velocity serif::discretization::blocks::rotation::angular_velocity_term {} [static], [constexpr]

Static constexpr instance of the angular_velocity term. This is used to access the term in the discretization.

10.468.3.2 name

std::string_view serif::discretization::blocks::rotation::name = meta::type_name<rotation>() [static], [constexpr]

Compile time reflected name of the field.

The documentation for this struct was generated from the following file:

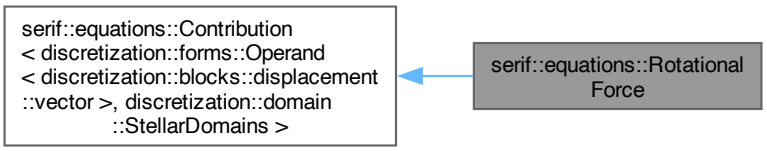
- src/include/serif/discretization/blocks/fields.hpp

10.469 serif::equations::RotationalForce Class Reference

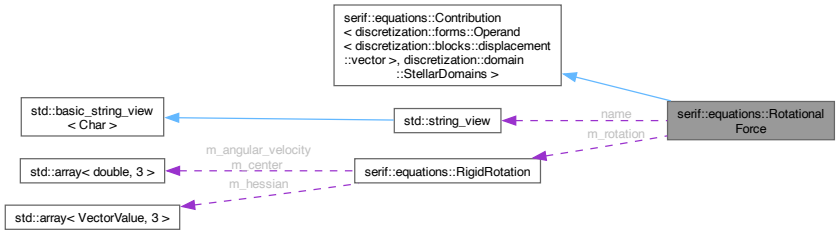
Rotational contribution to displacement balance: -integral_star rho grad(Psi) . w dV.

#include <rotational_force.hpp>

Inheritance diagram for serif::equations::RotationalForce:



Collaboration diagram for serif::equations::RotationalForce:



Public Types

- using Density = discretization::blocks::density::mass
- using AngularVelocity = discretization::blocks::rotation::angular_velocity
- using AngularVelocityValue = discretization::forms::Operand<AngularVelocity>
- using DensityValue = discretization::forms::Operand<Density>
- using Operands = utils::types::TypeList<DensityValue, AngularVelocityValue>
- using Result = discretization::forms::VectorValue

Public Types inherited from serif::equations::Contribution< discretization::forms::Operand< discretization::blocks::displacement::vecto

- using Test
- using TestOperand
- using Support

Public Member Functions

- RotationalForce (Result axis={0.0, 0.0, 1.0}, Result center={})

  Prescribe only the axis and center; angular speed is a global scalar operand.
- PointLinearization< Operands, Result > evaluate (const EquationPoint< Operands > &point) const

  F = -rho grad(Psi), dF/drho = -grad(Psi), dF/dx = -rho Hessian(Psi).
- int additional_order (DensityValue, int position_order) const
- int additional_order (AngularVelocityValue, int position_order) const

Static Public Attributes

- static constexpr std::string_view name = "rotational force"

Private Attributes

- RigidRotation m_rotation

10.469.1 Detailed Description

Rotational contribution to displacement balance: -integral_star rho grad(Psi) . w dV.

10.469.2 Member Typedef Documentation

10.469.2.1 AngularVelocity

using serif::equations::RotationalForce::AngularVelocity = discretization::blocks::rotation::angular_velocity

10.469.2.2 AngularVelocityValue

using serif::equations::RotationalForce::AngularVelocityValue = discretization::forms::Operand<AngularVelocity>

10.469.2.3 Density

using serif::equations::RotationalForce::Density = discretization::blocks::density::mass

10.469.2.4 DensityValue

using serif::equations::RotationalForce::DensityValue = discretization::forms::Operand<Density>

10.469.2.5 Operands

using serif::equations::RotationalForce::Operands = utils::types::TypeList<DensityValue, AngularVelocityValue>

10.469.2.6 Result

using serif::equations::RotationalForce::Result = discretization::forms::VectorValue

10.469.3 Constructor & Destructor Documentation

10.469.3.1 RotationalForce()

serif::equations::RotationalForce::RotationalForce (
Result axis = {0.0, 0.0, 1.0},
Result center = {}) [inline], [explicit]

Prescribe only the axis and center; angular speed is a global scalar operand.

10.469.4 Member Function Documentation

10.469.4.1 additional_order() [1/2]

int serif::equations::RotationalForce::additional_order (
AngularVelocityValue ,
int position_order) const [inline], [nodiscard]

10.469.4.2 additional_order() [2/2]

int serif::equations::RotationalForce::additional_order (
DensityValue ,
int position_order) const [inline], [nodiscard]

10.469.4.3 evaluate()

PointLinearization< Operands, Result > serif::equations::RotationalForce::evaluate (
const EquationPoint< Operands > & point) const [inline], [nodiscard]

F = -rho grad(Psi), dF/drho = -grad(Psi), dF/dx = -rho Hessian(Psi).

10.469.5 Member Data Documentation

10.469.5.1 m_rotation

RigidRotation serif::equations::RotationalForce::m_rotation [private]

10.469.5.2 name

std::string_view serif::equations::RotationalForce::name = "rotational force" [static], [constexpr]

The documentation for this class was generated from the following file:

- src/include/serif/equations/rotational_force.hpp

10.470 serif::discretization::quadrature::RuleControl Struct Reference

#include <rules.hpp>

Public Attributes

- std::optional< int > fixed_order
- int boost = 0

10.470.1 Member Data Documentation

10.470.1.1 boost

int serif::discretization::quadrature::RuleControl::boost = 0

10.470.1.2 fixed_order

std::optional<int> serif::discretization::quadrature::RuleControl::fixed_order

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

10.471 serif::discretization::quadrature::RuleSet Class Reference

#include <rules.hpp>

Collaboration diagram for serif::discretization::quadrature::RuleSet:



Public Member Functions

- template<forms::Form FormT>
  void set_form_control (const RuleControl control)
- void set_form_control (const forms::FormDescriptor &form, const RuleControl control)
- const RuleControl * find_form_control (const forms::FormDescriptor &form) const noexcept
- void set_role_control (const QuadratureRole role, const RuleControl control) noexcept
- const RuleControl & role_control (const QuadratureRole role) const noexcept
- void set_fallback_control (const RuleControl control) noexcept
- const RuleControl & fallback_control () const noexcept
- auto begin ()
- auto end ()
- auto begin () const
- auto end () const

Private Attributes

- std::vector< FormRule > m_formRules
- RuleControl m_discretizationControl
- RuleControl m_preconditionerControl
- RuleControl m_diagnosticControl
- RuleControl m_projectionControl
- RuleControl m_fallbackControl

10.471.1 Member Function Documentation

10.471.1.1 begin() [1/2]

auto serif::discretization::quadrature::RuleSet::begin () [inline]

10.471.1.2 begin() [2/2]

auto serif::discretization::quadrature::RuleSet::begin () const [inline]

10.471.1.3 end() [1/2]

auto serif::discretization::quadrature::RuleSet::end () [inline]

10.471.1.4 end() [2/2]

auto serif::discretization::quadrature::RuleSet::end () const [inline]

10.471.1.5 fallback_control()

const RuleControl & serif::discretization::quadrature::RuleSet::fallback_control () const [inline], [nodiscard], [noexcept]

10.471.1.6 find_form_control()

const RuleControl * serif::discretization::quadrature::RuleSet::find_form_control (
const forms::FormDescriptor & form) const [inline], [nodiscard], [noexcept]

10.471.1.7 role_control()

const RuleControl & serif::discretization::quadrature::RuleSet::role_control (
const QuadratureRole role) const [inline], [nodiscard], [noexcept]

10.471.1.8 set_fallback_control()

void serif::discretization::quadrature::RuleSet::set_fallback_control (
const RuleControl control) [inline], [noexcept]

10.471.1.9 set_form_control() [1/2]

void serif::discretization::quadrature::RuleSet::set_form_control (
const forms::FormDescriptor & form,
const RuleControl control) [inline]

10.471.1.10 set_form_control() [2/2]

template<forms::Form FormT>
void serif::discretization::quadrature::RuleSet::set_form_control (
const RuleControl control) [inline]

10.471.1.11 set_role_control()

void serif::discretization::quadrature::RuleSet::set_role_control (
const QuadratureRole role,
const RuleControl control) [inline], [noexcept]

10.471.2 Member Data Documentation

10.471.2.1 m_diagnosticControl

RuleControl serif::discretization::quadrature::RuleSet::m_diagnosticControl [private]

10.471.2.2 m_discretizationControl

RuleControl serif::discretization::quadrature::RuleSet::m_discretizationControl [private]

10.471.2.3 m_fallbackControl

RuleControl serif::discretization::quadrature::RuleSet::m_fallbackControl [private]

10.471.2.4 m_formRules

std::vector<FormRule> serif::discretization::quadrature::RuleSet::m_formRules [private]

10.471.2.5 m_preconditionerControl

RuleControl serif::discretization::quadrature::RuleSet::m_preconditionerControl [private]

10.471.2.6 m_projectionControl

RuleControl serif::discretization::quadrature::RuleSet::m_projectionControl [private]

The documentation for this class was generated from the following file:

- src/include/serif/discretization/quadrature/rules.hpp

10.472 serif::eos::runtime::RuntimeCatalogDispatch< EOS, CatalogT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

10.473 serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > > Struct Template Reference

#include <details.hpp>

Static Public Member Functions

- static std::expected< double, utils::errors::SERiFErrorCode > evaluate (const void *object, const dimensions::runtime::ThermodynamicQuantityID &outputQuantity, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues)
- static std::expected< double, utils::errors::SERiFErrorCode > partial_derivative (const void *object, const dimensions::runtime::ThermodynamicQuantityID &outputQuantity, const dimensions::runtime::ThermodynamicQ &withRespectTo, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues)

10.473.1 Member Function Documentation

10.473.1.1 evaluate()

template<typename EOS, typename... RelationTs>
static std::expected< double, utils::errors::SERiFErrorCode > serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > >::evaluate (
const void * object,
const dimensions::runtime::ThermodynamicQuantityID & outputQuantity,
const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues) [inline], [static], [nodiscard]

10.473.1.2 partial_derivative()

template<typename EOS, typename... RelationTs>
static std::expected< double, utils::errors::SERiFErrorCode > serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > >::partial_derivative (
const void * object,
const dimensions::runtime::ThermodynamicQuantityID & outputQuantity,
const dimensions::runtime::ThermodynamicQuantityID & withRespectTo,
const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues) [inline], [static], [nodiscard]

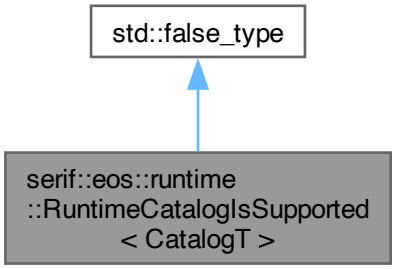
The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

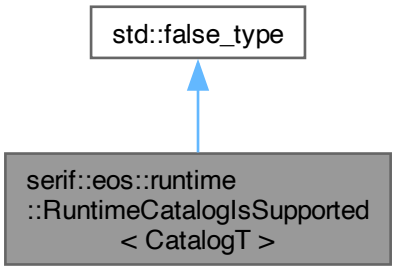
10.474 serif::eos::runtime::RuntimeCatalogIsSupported< CatalogT > Struct Template Reference

#include <details.hpp>

Inheritance diagram for serif::eos::runtime::RuntimeCatalogIsSupported< CatalogT >:



Collaboration diagram for serif::eos::runtime::RuntimeCatalogIsSupported< CatalogT >:



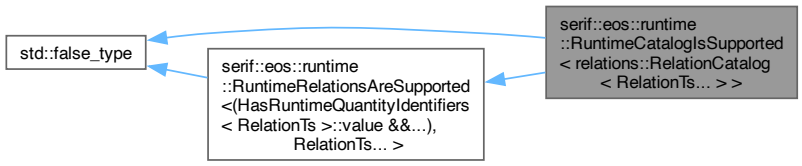
The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

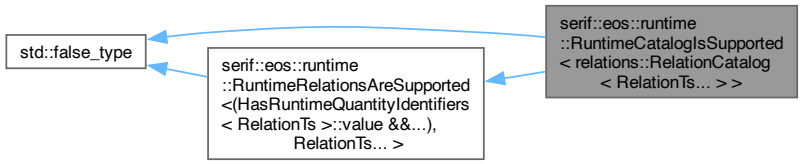
10.475 serif::eos::runtime::RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > > Struct Template Reference

#include <details.hpp>

Inheritance diagram for serif::eos::runtime::RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > >:



Collaboration diagram for serif::eos::runtime::RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > >:



The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

10.476 serif::eos::runtime::RuntimeCatalogStorage< EOS, CatalogT > Struct Template Reference

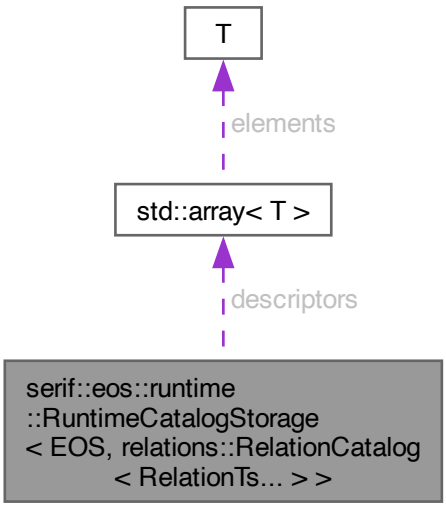
The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

10.477 serif::eos::runtime::RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > > Struct Template Reference

#include <details.hpp>

Collaboration diagram for serif::eos::runtime::RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > >:



Static Public Attributes

- static constexpr std::array descriptors

10.477.1 Member Data Documentation

10.477.1.1 descriptors

template<typename EOS, typename... RelationTs>
std::array serif::eos::runtime::RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > >::descriptors [inline], [static], [constexpr]

Initial value:

```
{
        RuntimeRelationStorage<EOS, RelationTs>::descriptor...
    }
```

The documentation for this struct was generated from the following file:

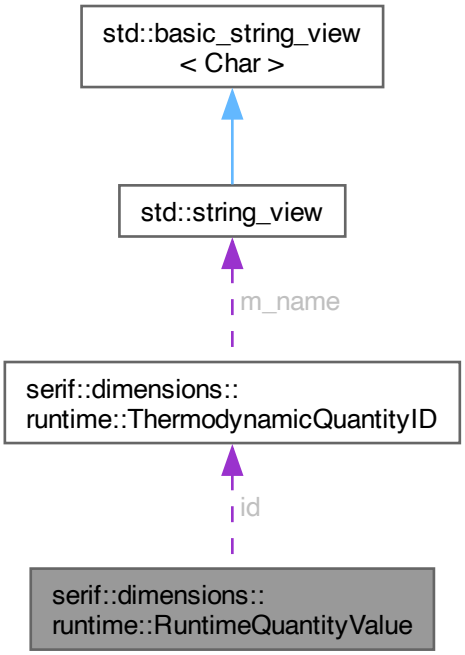
- src/include/serif/eos/runtime/details.hpp

10.478 serif::dimensions::runtime::RuntimeQuantityValue Struct Reference

Runtime resolved type for a quantity value. This holds both an ID and a value. This is the runtime equivalent to a QuantityValue. Note that where these differ is that QuantityValue is built with template types. This allows the compiler to resolve it to a plain numeric type and still do optimizations That is generally not the case with RuntimeQuantityValue.

#include <runtime.hpp>

Collaboration diagram for serif::dimensions::runtime::RuntimeQuantityValue:



Public Attributes

- ThermodynamicQuantityID id
- double value {}

10.478.1 Detailed Description

Runtime resolved type for a quantity value. This holds both an ID and a value. This is the runtime equivalent to a QuantityValue. Note that where these differ is that QuantityValue is built with template types. This allows the compiler to resolve it to a plain numeric type and still do optimizations That is generally not the case with RuntimeQuantityValue.

10.478.2 Member Data Documentation

10.478.2.1 id

ThermodynamicQuantityID serif::dimensions::runtime::RuntimeQuantityValue::id

10.478.2.2 value

double serif::dimensions::runtime::RuntimeQuantityValue::value {}

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/runtime/runtime.hpp

10.479 serif::dimensions::runtime::RuntimeRelationDescriptor Struct Reference

#include <runtime.hpp>

Collaboration diagram for serif::dimensions::runtime::RuntimeRelationDescriptor:



Public Member Functions

- constexpr bool hasPartialDerivative (const std::size_t inputIndex) const noexcept

Public Attributes

- ThermodynamicQuantityID outputQuantity
- std::span< const ThermodynamicQuantityID > inputQuantities
- std::uint64_t partialDerivativeMask

10.479.1 Member Function Documentation

10.479.1.1 hasPartialDerivative()

bool serif::dimensions::runtime::RuntimeRelationDescriptor::hasPartialDerivative (
const std::size_t inputIndex) const [inline], [nodiscard], [constexpr], [noexcept]

10.479.2 Member Data Documentation

10.479.2.1 inputQuantities

std::span<const ThermodynamicQuantityID> serif::dimensions::runtime::RuntimeRelationDescriptor::inputQuantities

10.479.2.2 outputQuantity

ThermodynamicQuantityID serif::dimensions::runtime::RuntimeRelationDescriptor::outputQuantity

10.479.2.3 partialDerivativeMask

std::uint64_t serif::dimensions::runtime::RuntimeRelationDescriptor::partialDerivativeMask

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/runtime/runtime.hpp

10.480 serif::eos::runtime::RuntimeRelationQuantities< RelationT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

10.481 serif::eos::runtime::RuntimeRelationQuantities< relations::Relation< OutputT, InputTs... > > Struct Template Reference

#include <details.hpp>

Public Types

- using Type = std::tuple<OutputT, InputTs...>

10.481.1 Member Typedef Documentation

10.481.1.1 Type

template<typename OutputT, typename... InputTs>
using serif::eos::runtime::RuntimeRelationQuantities< relations::Relation< OutputT, InputTs... > >::Type = std::tuple<OutputT, InputTs...>

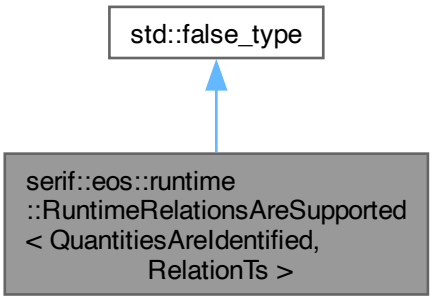
The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

10.482 serif::eos::runtime::RuntimeRelationsAreSupported< QuantitiesAreIdentified, RelationTs > Struct Template Reference

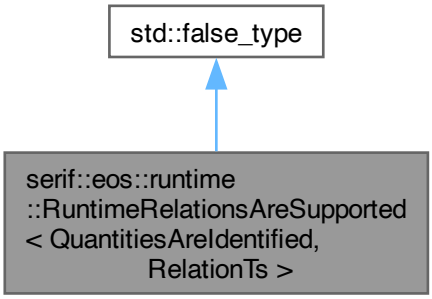
#include <details.hpp>

Inheritance diagram for serif::eos::runtime::RuntimeRelationsAreSupported< QuantitiesAreIdentified, RelationTs >:



Collaboration diagram for serif::eos::runtime::RuntimeRelationsAreSupported< QuantitiesAreIdentified,

RelationTs >:



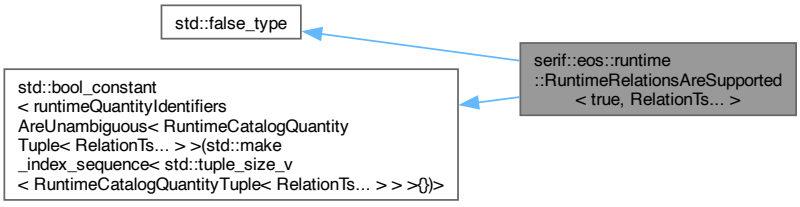
The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

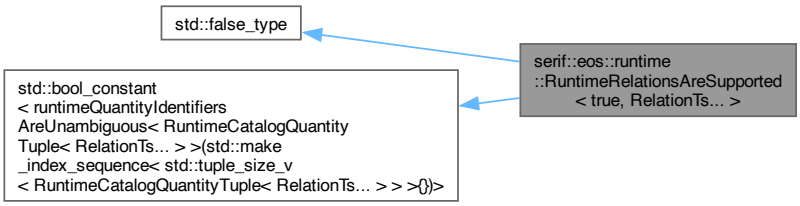
10.483 serif::eos::runtime::RuntimeRelationsAreSupported< true, RelationTs... > Struct Template Reference

#include <details.hpp>

Inheritance diagram for serif::eos::runtime::RuntimeRelationsAreSupported< true, RelationTs... >:



Collaboration diagram for serif::eos::runtime::RuntimeRelationsAreSupported< true, RelationTs... >:



The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

10.484 serif::eos::runtime::RuntimeRelationStorage< EOS, RelationT > Struct Template Reference

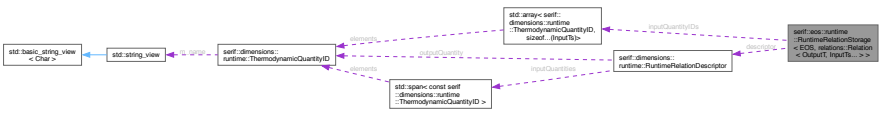
The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

10.485 serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > > Struct Template Reference

#include <details.hpp>

Collaboration diagram for serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >:



Public Types

- using RelationType = relations::Relation<OutputT, InputTs...>

Static Public Member Functions

- template<std::size_t... Indices>
  static consteval std::uint64_t makePartialDerivativeMask (std::index_sequence< Indices... >)

Static Public Attributes

- static constexpr std::array< dimensions::runtime::ThermodynamicQuantityID, sizeof...(InputTs)> inputQuantityIDs
- static constexpr std::uint64_t partialDerivativeMask = makePartialDerivativeMask(std::index_sequence_for<InputTs...>{})
- static constexpr dimensions::runtime::RuntimeRelationDescriptor descriptor

10.485.1 Member Typedef Documentation

10.485.1.1 RelationType

template<typename EOS, typename OutputT, typename... InputTs>
using serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >::RelationType = relations::Relation<OutputT, InputTs...>

10.485.2 Member Function Documentation

10.485.2.1 makePartialDerivativeMask()

template<typename EOS, typename OutputT, typename... InputTs>
template<std::size_t... Indices>
static consteval std::uint64_t serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >::makePartialDerivativeMask (
std::index_sequence< Indices... >) [inline], [static], [nodiscard], [consteval]

10.485.3 Member Data Documentation

10.485.3.1 descriptor

template<typename EOS, typename OutputT, typename... InputTs>
dimensions::runtime::RuntimeRelationDescriptor serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >::descriptor [inline], [static], [constexpr]

Initial value:

```
{
        .outputQuantity = dimensions::runtime::thermodynamicQuantityID<OutputT>,
        .inputQuantities = std::span<const dimensions::runtime::ThermodynamicQuantityID>{inputQuantityIDs},
        .partialDerivativeMask = partialDerivativeMask
    }
```

10.485.3.2 inputQuantityIDs

template<typename EOS, typename OutputT, typename... InputTs>
std::array<dimensions::runtime::ThermodynamicQuantityID, sizeof...(InputTs)> serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >::inputQuantityIDs [inline], [static], [constexpr]

Initial value:

```
{
        dimensions::runtime::thermodynamicQuantityID<InputTs>...
    }
```

10.485.3.3 partialDerivativeMask

template<typename EOS, typename OutputT, typename... InputTs>
std::uint64_t serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >::partialDerivativeMask = makePartialDerivativeMask(std::index_sequence_for<InputTs...>{}) [inline], [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/eos/runtime/details.hpp

10.486 serif::utils::dependencies::keys::Scale Struct Reference

A multiplicative scale supplied as an input.

#include <keys.hpp>

10.486.1 Detailed Description

A multiplicative scale supplied as an input.

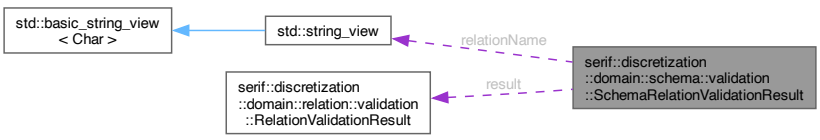
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/keys.hpp

10.487 serif::discretization::domain::schema::validation::SchemaRelationValidationResult Struct Reference

#include <results.hpp>

Collaboration diagram for serif::discretization::domain::schema::validation::SchemaRelationValidationResult:



Public Member Functions

- bool valid () const noexcept
- operator bool () const noexcept

Public Attributes

- std::size_t relationIndex {0}
- std::string_view relationName
- relation::validation::RelationValidationResult result

10.487.1 Member Function Documentation

10.487.1.1 operator bool()

serif::discretization::domain::schema::validation::SchemaRelationValidationResult::operator bool () const [explicit], [nodiscard], [noexcept]

10.487.1.2 valid()

bool serif::discretization::domain::schema::validation::SchemaRelationValidationResult::valid () const [nodiscard], [noexcept]

10.487.2 Member Data Documentation

10.487.2.1 relationIndex

std::size_t serif::discretization::domain::schema::validation::SchemaRelationValidationResult::relationIndex {0}

10.487.2.2 relationName

std::string_view serif::discretization::domain::schema::validation::SchemaRelationValidationResult::relationName

10.487.2.3 result

relation::validation::RelationValidationResult serif::discretization::domain::schema::validation::SchemaRelationValidation↩
Result::result

The documentation for this struct was generated from the following files:

- src/include/serif/discretization/domain/schema/validation/results.hpp
- src/lib/serif/discretization/domain/schema/validation/results.cpp

10.488 serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, RelationListT > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/schema/validation/validator.hpp

10.489 serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, relation::lists::RelationList< RelationTs... > > Struct Template Reference

#include <validator.hpp>

Static Public Member Functions

- static SchemaValidationResult validate (const mesh::MeshTopology &topology)

10.489.1 Member Function Documentation

10.489.1.1 validate()

template<IsSchema SchemaT, IsRelation... RelationTs>
static SchemaValidationResult serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, relation::lists::RelationList< RelationTs... > >::validate (
const mesh::MeshTopology & topology) [inline], [static], [nodiscard]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/schema/validation/validator.hpp

10.490 serif::discretization::domain::schema::validation::SchemaValidationResult Struct Reference

#include <results.hpp>

Collaboration diagram for serif::discretization::domain::schema::validation::SchemaValidationResult:



Public Member Functions

- bool valid () const noexcept
- operator bool () const noexcept
- std::size_t relation_count () const noexcept
- std::size_t failed_relation_count () const noexcept
- std::size_t passed_relation_count () const noexcept
- std::optional< std::size_t > first_failed_relation_index () const noexcept

Public Attributes

- std::vector< SchemaRelationValidationResult > relationResults

10.490.1 Member Function Documentation

10.490.1.1 failed_relation_count()

std::size_t serif::discretization::domain::schema::validation::SchemaValidationResult::failed_relation_count () const [nodiscard], [noexcept]

10.490.1.2 first_failed_relation_index()

std::optional< std::size_t > serif::discretization::domain::schema::validation::SchemaValidationResult::first_failed_relation_index () const [nodiscard], [noexcept]

10.490.1.3 operator bool()

serif::discretization::domain::schema::validation::SchemaValidationResult::operator bool () const [explicit], [nodiscard], [noexcept]

10.490.1.4 passed_relation_count()

std::size_t serif::discretization::domain::schema::validation::SchemaValidationResult::passed_relation_count () const [nodiscard], [noexcept]

10.490.1.5 relation_count()

std::size_t serif::discretization::domain::schema::validation::SchemaValidationResult::relation_count () const [nodiscard], [noexcept]

10.490.1.6 valid()

bool serif::discretization::domain::schema::validation::SchemaValidationResult::valid () const [nodiscard], [noexcept]

10.490.2 Member Data Documentation

10.490.2.1 relationResults

std::vector<SchemaRelationValidationResult> serif::discretization::domain::schema::validation::SchemaValidationResult↩
::relationResults

The documentation for this struct was generated from the following files:

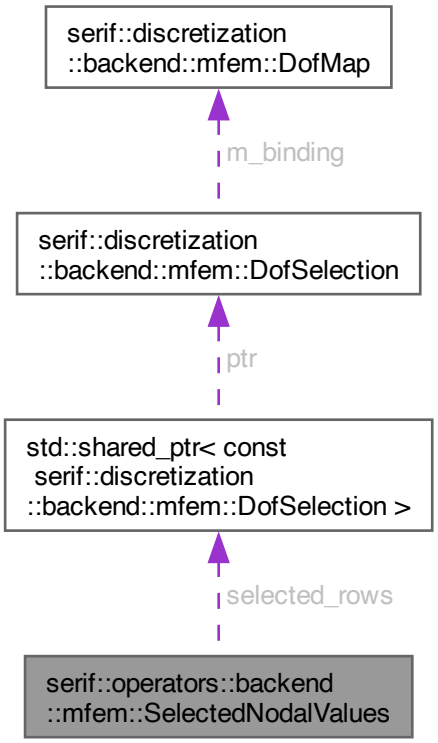
- src/include/serif/discretization/domain/schema/validation/results.hpp
- src/lib/serif/discretization/domain/schema/validation/results.cpp

10.491 serif::operators::backend::mfem::SelectedNodalValues Struct Reference

Selected coefficients are owned by the context; the selection has a stable shared identity.

#include <nodal_inputs.hpp>

Collaboration diagram for serif::operators::backend::mfem::SelectedNodalValues:



Public Attributes

- std::shared_ptr< const discretization::backend::mfem::DofSelection > selected_rows
- ::mfem::Vector values

10.491.1 Detailed Description

Selected coefficients are owned by the context; the selection has a stable shared identity.

10.491.2 Member Data Documentation

10.491.2.1 selected_rows

std::shared_ptr<const discretization::backend::mfem::DofSelection> serif::operators::backend::mfem::SelectedNodalValues↩
::selected_rows

10.491.2.2 values

::mfem::Vector serif::operators::backend::mfem::SelectedNodalValues::values

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/nodal_inputs.hpp

10.492 serif::utils::errors::serif_error_printer< Message, Code > Struct Template Reference

The documentation for this struct was generated from the following file:

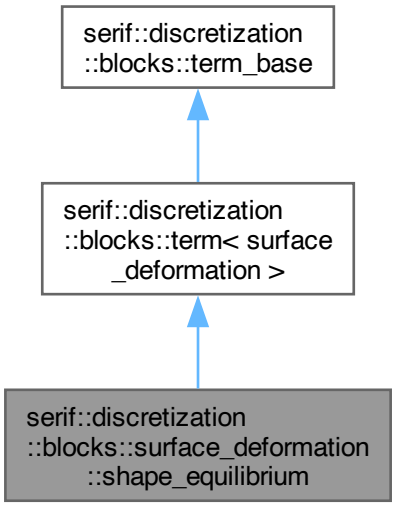
- src/include/serif/utils/error/errors.hpp

10.493 serif::discretization::blocks::surface_deformation::shape_↩ equilibrium Struct Reference

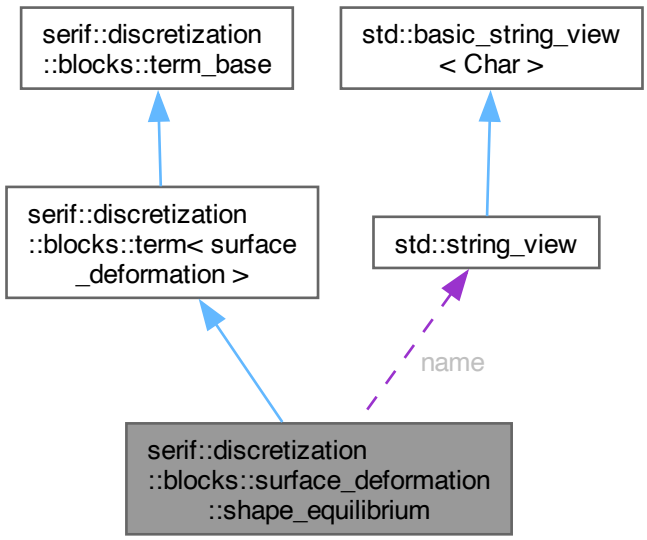
The shape equilibrium term represents the shape equilibrium condition of the system. This is a term that is associated with the surface deformation field and is used to enforce the shape equilibrium condition in the weak form. This is where we associate the residuals with. Note how we do not have a functional space, support, or a quantity. This is because the shape equilibrium condition is a global condition that is not associated with any particular functional space, support, or quantity.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::surface_deformation::shape_equilibrium:



Collaboration diagram for serif::discretization::blocks::surface_deformation::shape_equilibrium:



Classes

- struct residual

  The residuals of the shape equilibrium condition.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<shape_equilibrium>()

  Compile time reflected name of the term.

Additional Inherited Members

Public Types inherited from serif::discretization::blocks::term< surface_deformation >

- using field_type

10.493.1 Detailed Description

The shape equilibrium term represents the shape equilibrium condition of the system. This is a term that is associated with the surface deformation field and is used to enforce the shape equilibrium condition in the weak form. This is where we associate the residuals with. Note how we do not have a functional space, support, or a quantity. This is because the shape equilibrium condition is a global condition that is not associated with any particular functional space, support, or quantity.

10.493.2 Member Data Documentation

10.493.2.1 name

std::string_view serif::discretization::blocks::surface_deformation::shape_equilibrium::name = meta::type_name<shape_equilibrium>() [static], [constexpr]

Compile time reflected name of the term.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

10.494 serif::geometry::deformation::keys::ShapeCoefficients Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

10.495 serif::geometry::deformation::spaces::ShapeParameters Struct Reference

Independent coordinates belonging to the selected surface representation.

#include <spaces.hpp>

10.495.1 Detailed Description

Independent coordinates belonging to the selected surface representation.

The documentation for this struct was generated from the following file:

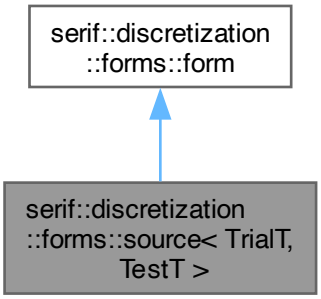
- src/include/serif/geometry/deformation/spaces.hpp

10.496 serif::discretization::forms::source< TrialT, TestT > Struct Template Reference

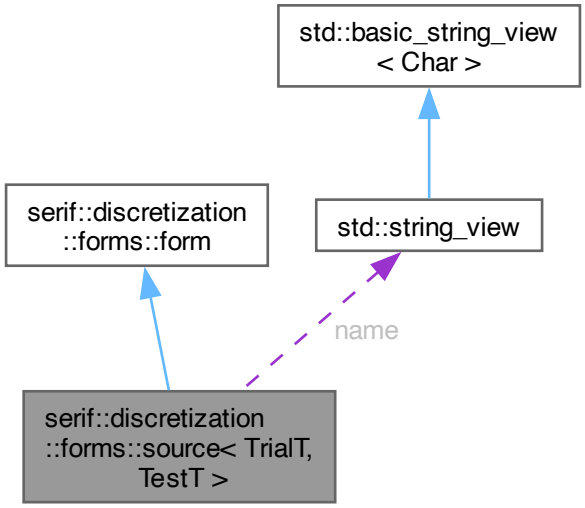
The source form represents the source term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the source operation.

#include <forms.hpp>

Inheritance diagram for serif::discretization::forms::source< TrialT, TestT >:



Collaboration diagram for serif::discretization::forms::source< TrialT, TestT >:



Public Types

- using trial_type = TrialT

  The trial term type for the source form.
- using test_type = TestT

  The test term type for the source form.
- using operand_types = utils::types::TypeList<Operand<TrialT>, Operand<TestT>>

  Type list of operand types for the source form, used for compile-time reflection and iteration over the operands.

Static Public Attributes

- static constexpr std::string_view name = meta::template_name<source>()

  Compile time reflected name of the form.

10.496.1 Detailed Description

template<blocks::Term TrialT, blocks::Term TestT>
struct serif::discretization::forms::source< TrialT, TestT >

The source form represents the source term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the source operation.

Template Parameters

TrialT	The trial term type.
TestT	The test term type.

There is a similar story here to the mass matrix, though this is much closer to what we expect in physics for a source term.

10.496.2 Member Typedef Documentation

10.496.2.1 operand_types

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::source< TrialT, TestT >::operand_types = utils::types::TypeList<Operand<TrialT>, Operand<TestT>>

Type list of operand types for the source form, used for compile-time reflection and iteration over the operands.

10.496.2.2 test_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::source< TrialT, TestT >::test_type = TestT

The test term type for the source form.

10.496.2.3 trial_type

template<blocks::Term TrialT, blocks::Term TestT>
using serif::discretization::forms::source< TrialT, TestT >::trial_type = TrialT

The trial term type for the source form.

10.496.3 Member Data Documentation

10.496.3.1 name

template<blocks::Term TrialT, blocks::Term TestT>
std::string_view serif::discretization::forms::source< TrialT, TestT >::name = meta::template_name<source>() [static], [constexpr]

Compile time reflected name of the form.

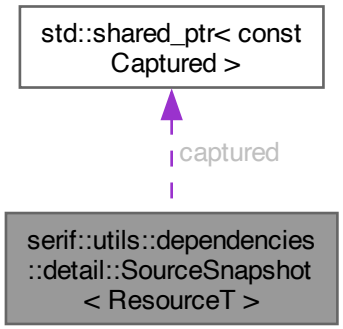
The documentation for this struct was generated from the following file:

- src/include/serif/discretization/forms/forms.hpp

10.497 serif::utils::dependencies::detail::SourceSnapshot< ResourceT > Struct Template Reference

#include <cache_storage.hpp>

Collaboration diagram for serif::utils::dependencies::detail::SourceSnapshot< ResourceT >:



Public Types

- using Captured = decltype(std::declval<const typename ResourceT::Policy&>().capture(std::declval<const typename ResourceT::Target&>()))

Public Attributes

- const void * identity {nullptr}
- std::shared_ptr< const Captured > captured

10.497.1 Member Typedef Documentation

10.497.1.1 Captured

template<typename ResourceT>
using serif::utils::dependencies::detail::SourceSnapshot< ResourceT >::Captured = decltype(std::declval<const typename ResourceT::Policy&>().capture(std::declval<const typename ResourceT::Target&>()))

10.497.2 Member Data Documentation

10.497.2.1 captured

template<typename ResourceT>
std::shared_ptr<const Captured> serif::utils::dependencies::detail::SourceSnapshot< ResourceT >::captured

10.497.2.2 identity

template<typename ResourceT>
const void* serif::utils::dependencies::detail::SourceSnapshot< ResourceT >::identity {nullptr}

The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_storage.hpp

10.498 serif::discretization::functional_space::SpaceDescriptor Struct Reference

Compile-time descriptor for a functional space. This struct is used to describe the family, continuity, trace, and mapping of a functional space at compile time.

#include <space.hpp>

Public Attributes

- const Family family

  The family of the functional space (e.g. L2, H1, H(div), H(curl)).
- const Continuity continuity

  The continuity of the functional space (e.g. discontinuous, continuous, normal continuous, tangential continuous).
- const Trace trace

  The trace of the functional space (e.g. none, value, normal, tangential).
- const Mapping mapping

  The mapping of the functional space (e.g. identity, contravariant_piola, covariant_piola).

10.498.1 Detailed Description

Compile-time descriptor for a functional space. This struct is used to describe the family, continuity, trace, and mapping of a functional space at compile time.

10.498.2 Member Data Documentation

10.498.2.1 continuity

const Continuity serif::discretization::functional_space::SpaceDescriptor::continuity

The continuity of the functional space (e.g. discontinuous, continuous, normal continuous, tangential continuous).

10.498.2.2 family

const Family serif::discretization::functional_space::SpaceDescriptor::family

The family of the functional space (e.g. L2, H1, H(div), H(curl)).

10.498.2.3 mapping

const Mapping serif::discretization::functional_space::SpaceDescriptor::mapping

The mapping of the functional space (e.g. identity, contravariant_piola, covariant_piola).

10.498.2.4 trace

const Trace serif::discretization::functional_space::SpaceDescriptor::trace

The trace of the functional space (e.g. none, value, normal, tangential).

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/functional_space/space.hpp

10.499 serif::utils::types::SparseMatrixBase< T > Struct Template Reference

#include <tensor.hpp>

Public Types

- using Type = T

10.499.1 Member Typedef Documentation

10.499.1.1 Type

template<typename T>
using serif::utils::types::SparseMatrixBase< T >::Type = T

The documentation for this struct was generated from the following file:

- src/include/serif/utils/types/tensor/tensor.hpp

10.500 serif::utils::types::SparseTensorBase< T, Rank > Struct Template Reference

#include <tensor.hpp>

Public Types

- using Type = T

Static Public Attributes

- static constexpr std::size_t rank = Rank

10.500.1 Member Typedef Documentation

10.500.1.1 Type

template<typename T, std::size_t Rank>
using serif::utils::types::SparseTensorBase< T, Rank >::Type = T

10.500.2 Member Data Documentation

10.500.2.1 rank

template<typename T, std::size_t Rank>
std::size_t serif::utils::types::SparseTensorBase< T, Rank >::rank = Rank [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/utils/types/tensor/tensor.hpp

10.501 serif::discretization::blocks::SpatialTermTraits< TermT > Struct Template Reference

Helper struct representing the traits of some spatial term type, including its functional space, support, family, order, and rank.

#include <traits.hpp>

Public Types

- using FunctionalSpace = TermT::FunctionalSpace
  The functional space type associated with the term. e.g. 2nd order scalar H1 space.
- using Support = TermT::Support
  The support type associated with the term. e.g. stellar domain.

Static Public Attributes

- static constexpr auto family = FunctionalSpace::functional_space.family
  The family of the functional space associated with the term. e.g. H1, L2, etc.
- static constexpr int order = FunctionalSpace::order
  The order of the functional space associated with the term. e.g. 1st order, 2nd order, etc.
- static constexpr std::size_t rank = FunctionalSpace::rank::value
  The rank of the functional space associated with the term.

10.501.1 Detailed Description

template<SpatialTerm TermT>
struct serif::discretization::blocks::SpatialTermTraits< TermT >

Helper struct representing the traits of some spatial term type, including its functional space, support, family, order, and rank.

Template Parameters

TermT	The spatial term type from which to extract the functional space and support types.

10.501.2 Member Typedef Documentation

10.501.2.1 FunctionalSpace

template<SpatialTerm TermT>
using serif::discretization::blocks::SpatialTermTraits< TermT >::FunctionalSpace = TermT::FunctionalSpace

The functional space type associated with the term. e.g. 2nd order scalar H1 space.

10.501.2.2 Support

template<SpatialTerm TermT>
using serif::discretization::blocks::SpatialTermTraits< TermT >::Support = TermT::Support

The support type associated with the term. e.g. stellar domain.

10.501.3 Member Data Documentation

10.501.3.1 family

template<SpatialTerm TermT>
auto serif::discretization::blocks::SpatialTermTraits< TermT >::family = FunctionalSpace::functional_space.family [static], [constexpr]

The family of the functional space associated with the term. e.g. H1, L2, etc.

10.501.3.2 order

template<SpatialTerm TermT>
int serif::discretization::blocks::SpatialTermTraits< TermT >::order = FunctionalSpace::order [static], [constexpr]

The order of the functional space associated with the term. e.g. 1st order, 2nd order, etc.

10.501.3.3 rank

template<SpatialTerm TermT>
std::size_t serif::discretization::blocks::SpatialTermTraits< TermT >::rank = FunctionalSpace::rank::value [static], [constexpr]

The rank of the functional space associated with the term.

The documentation for this struct was generated from the following file:

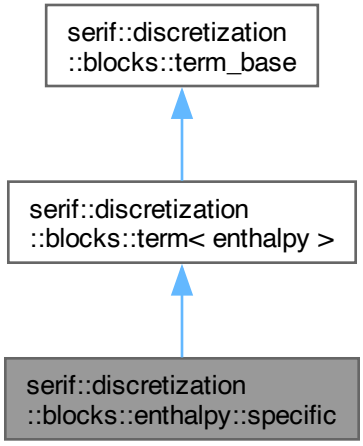
- src/include/serif/discretization/blocks/traits.hpp

10.502 serif::discretization::blocks::enthalpy::specific Struct Reference

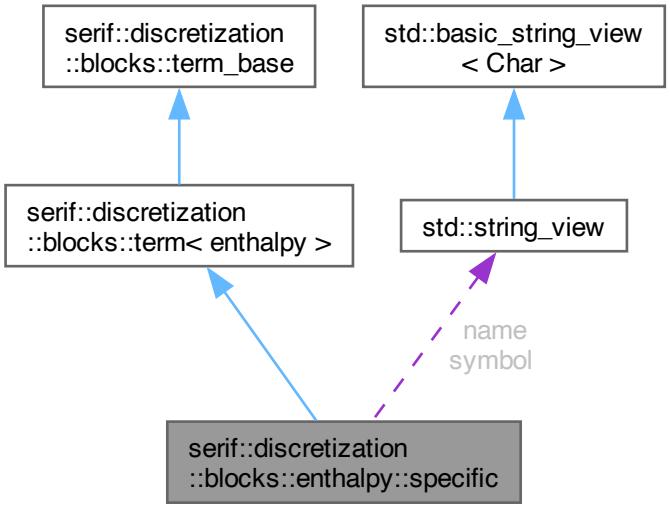
the specific enthalpy is the term we specifically work with rather than the total enthalpy of dof. In fact heuristically I would say that generally we should always work in specific quantities rather than total quantities as there is not a well defined amount of material at some dof.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::enthalpy::specific:



Collaboration diagram for serif::discretization::blocks::enthalpy::specific:



Classes

- struct residual

  The residual block is a block that contains the residual of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system.
- struct value

  The value block is a block that contains the value of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system.

Public Types

- using FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::H1, 3, 0>
- using Support = domain::StellarDomains

  There is only material in the stellar domain therefore we only support the enthalpy field in the stellar domain.
- using Quantity = dimensions::SpecificEnthalpy

  This has the dimension of specific enthalpy,.

Public Types inherited from serif::discretization::blocks::term< enthalpy >

- using field_type

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<specific>()
  Compile time reflected name of the term.
- static constexpr std::string_view symbol = "h"
  Compile time symbol used for the term.

10.502.1 Detailed Description

the specific enthalpy is the term we specifically work with rather than the total enthalpy of dof. In fact heuristically I would say that generally we should always work in specific quantities rather than total quantities as there is not a well defined amount of material at some dof.

10.502.2 Member Typedef Documentation

10.502.2.1 FunctionalSpace

using serif::discretization::blocks::enthalpy::specific::FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::H1, 3, 0>

10.502.2.2 Quantity

using serif::discretization::blocks::enthalpy::specific::Quantity = dimensions::SpecificEnthalpy

This has the dimension of specific enthalpy,.

10.502.2.3 Support

using serif::discretization::blocks::enthalpy::specific::Support = domain::StellarDomains

There is only material in the stellar domain therefore we only support the enthalpy field in the stellar domain.

10.502.3 Member Data Documentation

10.502.3.1 name

std::string_view serif::discretization::blocks::enthalpy::specific::name = meta::type_name<specific>() [static], [constexpr]

Compile time reflected name of the term.

10.502.3.2 symbol

std::string_view serif::discretization::blocks::enthalpy::specific::symbol = "h" [static], [constexpr]

Compile time symbol used for the term.

The documentation for this struct was generated from the following file:

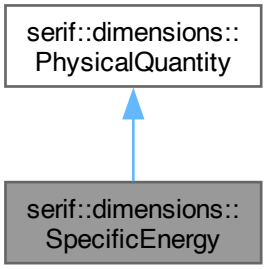
- src/include/serif/discretization/blocks/fields.hpp

10.503 serif::dimensions::SpecificEnergy Struct Reference

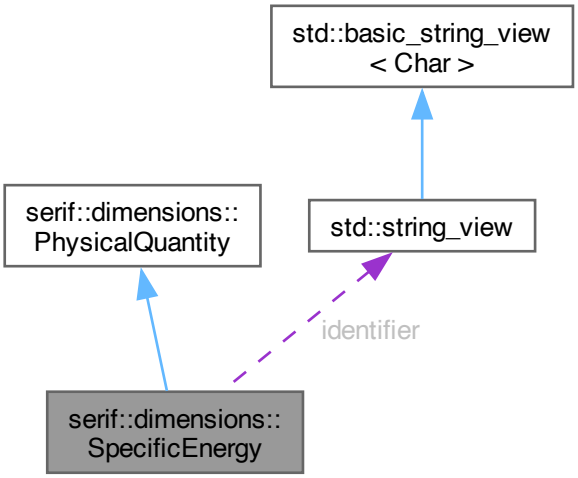
Dimension of Specific Energy.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::SpecificEnergy:



Collaboration diagram for serif::dimensions::SpecificEnergy:



Static Public Attributes

- static constexpr std::string_view identifier = "specific_energy"

10.503.1 Detailed Description

Dimension of Specific Energy.

10.503.2 Member Data Documentation

10.503.2.1 identifier

std::string_view serif::dimensions::SpecificEnergy::identifier = "specific_energy" [static], [constexpr]

The documentation for this struct was generated from the following file:

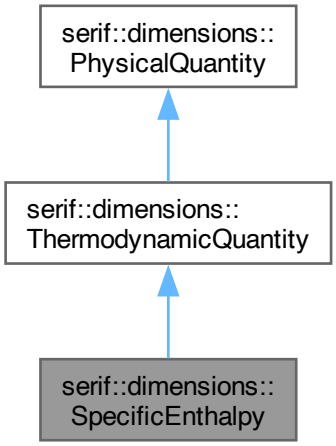
- src/include/serif/dimensions/dimensions.hpp

10.504 serif::dimensions::SpecificEnthalpy Struct Reference

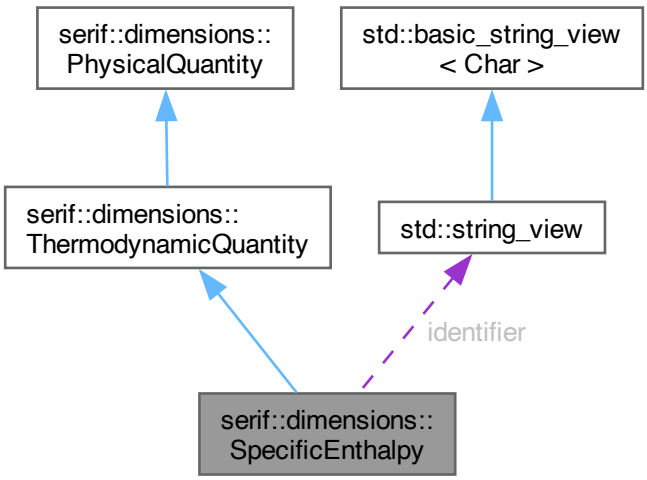
Dimension of Specific Enthalpy.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::SpecificEnthalpy:



Collaboration diagram for serif::dimensions::SpecificEnthalpy:



Static Public Attributes

- static constexpr std::string_view identifier = ”specific_enthalpy”

10.504.1 Detailed Description

Dimension of Specific Enthalpy.

10.504.2 Member Data Documentation

10.504.2.1 identifier

std::string_view serif::dimensions::SpecificEnthalpy::identifier = ”specific_enthalpy” [static], [constexpr]

The documentation for this struct was generated from the following file:

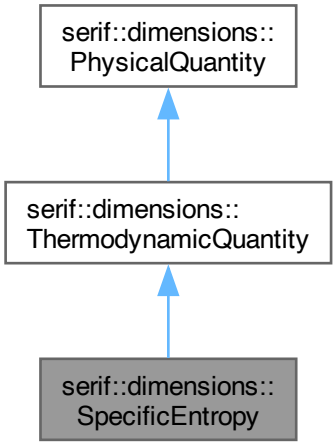
- src/include/serif/dimensions/dimensions.hpp

10.505 serif::dimensions::SpecificEntropy Struct Reference

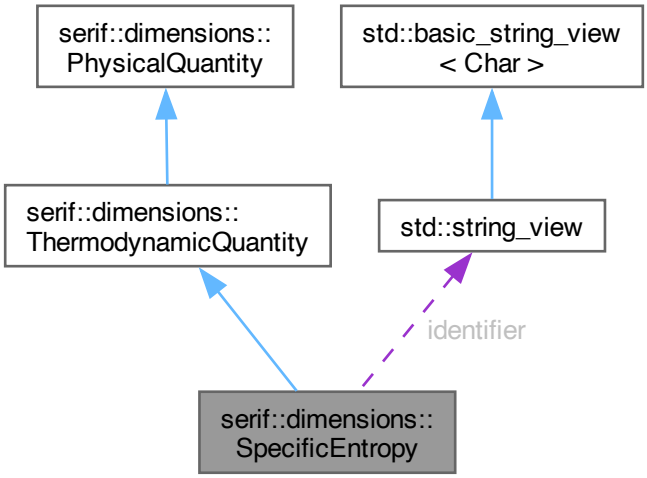
Dimension of Specific Entropy.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::SpecificEntropy:



Collaboration diagram for serif::dimensions::SpecificEntropy:



Static Public Attributes

- static constexpr std::string_view identifier = "specific_entropy"

10.505.1 Detailed Description

Dimension of Specific Entropy.

10.505.2 Member Data Documentation

10.505.2.1 identifier

std::string_view serif::dimensions::SpecificEntropy::identifier = "specific_entropy" [static], [constexpr]

The documentation for this struct was generated from the following file:

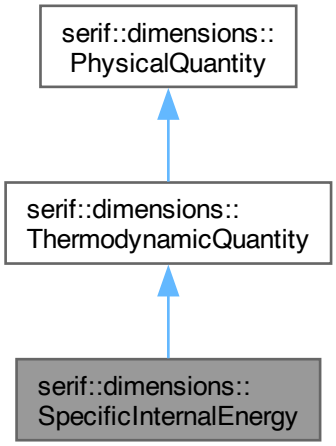
- src/include/serif/dimensions/dimensions.hpp

10.506 serif::dimensions::SpecificInternalEnergy Struct Reference

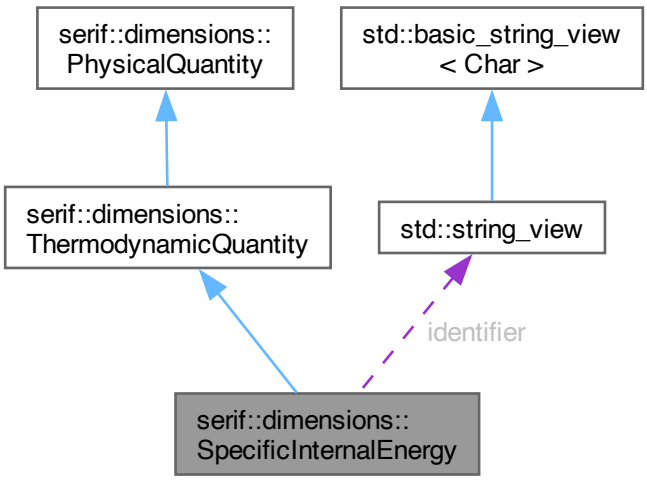
Dimension of Specific Internal Energy.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::SpecificInternalEnergy:



Collaboration diagram for serif::dimensions::SpecificInternalEnergy:



Static Public Attributes

- static constexpr std::string_view identifier = ”specific_internal_energy”

10.506.1 Detailed Description

Dimension of Specific Internal Energy.

10.506.2 Member Data Documentation

10.506.2.1 identifier

std::string_view serif::dimensions::SpecificInternalEnergy::identifier = ”specific_internal_energy” [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.507 serif::geometry::deformation::SphericalHarmonicMode Struct Reference

Identify a real harmonic, with negative orders using sine and positive orders using cosine.

#include <surface_embedding.hpp>

Public Member Functions

- bool operator== (const SphericalHarmonicMode &) const =default
  Mode identity includes the sign of the azimuthal order.

Public Attributes

- int degree
- int order

10.507.1 Detailed Description

Identify a real harmonic, with negative orders using sine and positive orders using cosine.

10.507.2 Member Function Documentation

10.507.2.1 operator==()

bool serif::geometry::deformation::SphericalHarmonicMode::operator== (
const SphericalHarmonicMode &) const [nodiscard], [default]

Mode identity includes the sign of the azimuthal order.

10.507.3 Member Data Documentation

10.507.3.1 degree

int serif::geometry::deformation::SphericalHarmonicMode::degree

10.507.3.2 order

int serif::geometry::deformation::SphericalHarmonicMode::order

The documentation for this struct was generated from the following file:

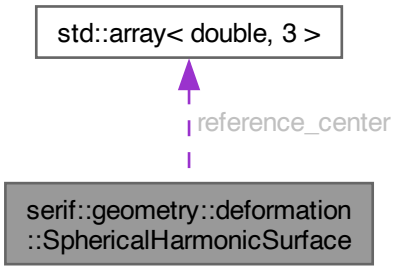
- src/include/serif/geometry/deformation/surface_embedding.hpp

10.508 serif::geometry::deformation::SphericalHarmonicSurface Struct Reference

Represent radial displacement lengths with normalized real spherical harmonics on the reference surface.

#include <models.hpp>

Collaboration diagram for serif::geometry::deformation::SphericalHarmonicSurface:



Public Member Functions

- bool operator== (const SphericalHarmonicSurface &) const =default

  A changed center or harmonic selection changes the physical representation.

Public Attributes

- std::array< double, 3 > reference_center {}
- int maximum_degree {4}
- std::optional< int > maximum_azimuthal_order {}

10.508.1 Detailed Description

Represent radial displacement lengths with normalized real spherical harmonics on the reference surface.

10.508.2 Member Function Documentation

10.508.2.1 operator==()

bool serif::geometry::deformation::SphericalHarmonicSurface::operator== (
const SphericalHarmonicSurface &) const [nodiscard], [default]

A changed center or harmonic selection changes the physical representation.

10.508.3 Member Data Documentation

10.508.3.1 maximum_azimuthal_order

std::optional<int> serif::geometry::deformation::SphericalHarmonicSurface::maximum_azimuthal_order {}

10.508.3.2 maximum_degree

int serif::geometry::deformation::SphericalHarmonicSurface::maximum_degree {4}

10.508.3.3 reference_center

std::array<double, 3> serif::geometry::deformation::SphericalHarmonicSurface::reference_center {}

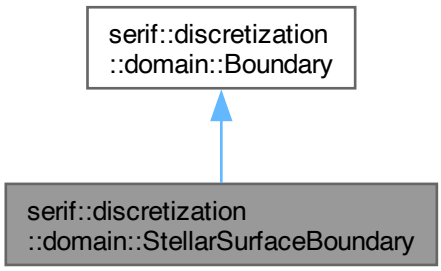
The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/models.hpp

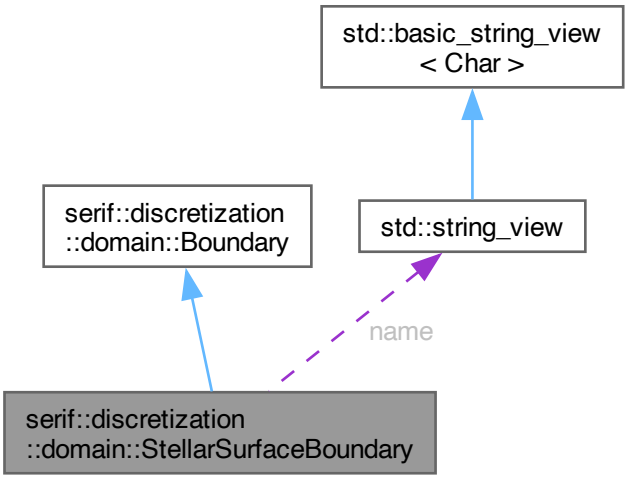
10.509 serif::discretization::domain::StellarSurfaceBoundary Struct Reference

#include <types.hpp>

Inheritance diagram for serif::discretization::domain::StellarSurfaceBoundary:



Collaboration diagram for serif::discretization::domain::StellarSurfaceBoundary:



Static Public Attributes

- static constexpr std::string_view name = "stellar_surface"

10.509.1 Member Data Documentation

10.509.1.1 name

std::string_view serif::discretization::domain::StellarSurfaceBoundary::name = ”stellar_surface” [static], [constexpr]

The documentation for this struct was generated from the following file:

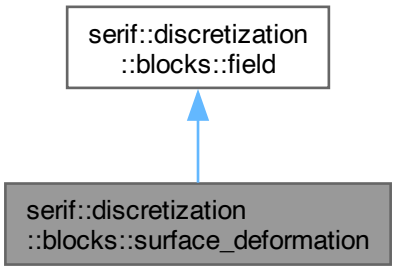
- src/include/serif/discretization/domain/types.hpp

10.510 serif::discretization::blocks::surface_deformation Struct Reference

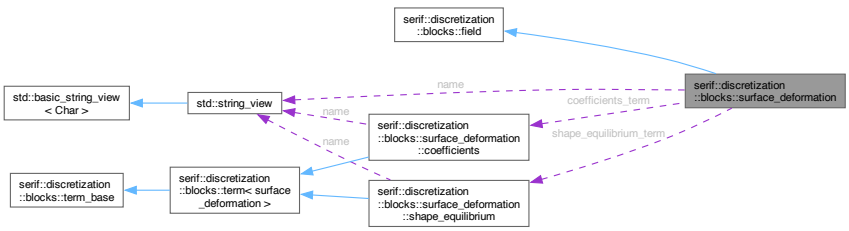
The surface corollary field to the full displacement field. This is in some ways more general than the others in so far as it is not pointwise defined over the domain. Rather, it is some general field of numbers, divorced from the underlying discrimination, that is used to represent the surface deformation of the system. This may be used, for example to represent spherical harmonic coefficients of the surface deformation. For more information see the surface and deformation systems.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::surface_deformation:



Collaboration diagram for serif::discretization::blocks::surface_deformation:



Classes

- struct coefficients

  The coefficients term represents some set of coefficients associated with whatever surface representation is being used to represent the surface deformation. Note how we do not have a functional space, support, or a quantity. This is also where we associate the values with.
- struct shape_equilibrium

  The shape equilibrium term represents the shape equilibrium condition of the system. This is a term that is associated with the surface deformation field and is used to enforce the shape equilibrium condition in the weak form. This is where we associate the residuals with. Note how we do not have a functional space, support, or a quantity. This is because the shape equilibrium condition is a global condition that is not associated with any particular functional space, support, or quantity.

Public Types

- using Terms = utils::types::TypeList<coefficients, shape_equilibrium>

  Type list of terms for compile time reflection and iteration over the terms of the field.

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<surface_deformation>()

  Compile time reflected name of the field.
- static constexpr coefficients coefficients_term {}

  Static constexpr instance of the coefficients term. This is used to access the term in the discretization.
- static constexpr shape_equilibrium shape_equilibrium_term {}

  Static constexpr instance of the shape_equilibrium term. This is used to access the term in the discretization.

10.510.1 Detailed Description

The surface corollary field to the full displacement field. This is in some ways more general than the others in so far as it is not pointwise defined over the domain. Rather, it is some general field of numbers, divorced from the underlying discrimination, that is used to represent the surface deformation of the system. This may be used, for example to represent spherical harmonic coefficients of the surface deformation. For more information see the surface and deformation systems.

10.510.2 Member Typedef Documentation

10.510.2.1 Terms

using serif::discretization::blocks::surface_deformation::Terms = utils::types::TypeList<coefficients, shape_equilibrium>

Type list of terms for compile time reflection and iteration over the terms of the field.

10.510.3 Member Data Documentation

10.510.3.1 coefficients_term

coefficients serif::discretization::blocks::surface_deformation::coefficients_term {} [static], [constexpr]

Static constexpr instance of the coefficients term. This is used to access the term in the discretization.

10.510.3.2 name

std::string_view serif::discretization::blocks::surface_deformation::name = meta::type_name<surface_deformation>() [static], [constexpr]

Compile time reflected name of the field.

10.510.3.3 shape_equilibrium_term

shape_equilibrium serif::discretization::blocks::surface_deformation::shape_equilibrium_term {} [static], [constexpr]

Static constexpr instance of the shape_equilibrium term. This is used to access the term in the discretization.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

10.511 serif::geometry::deformation::keys::SurfaceDisplacement Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

10.512 serif::geometry::deformation::spaces::SurfaceDisplacement Struct Reference

Cartesian surface traces with components interleaved at each canonical node.

#include <spaces.hpp>

10.512.1 Detailed Description

Cartesian surface traces with components interleaved at each canonical node.

The documentation for this struct was generated from the following file:

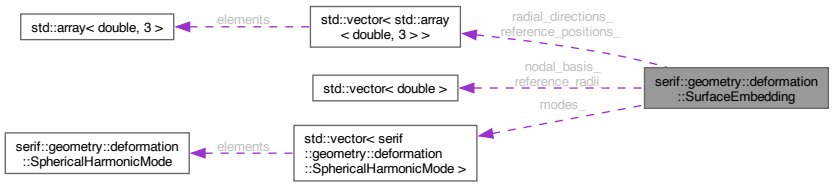
- src/include/serif/geometry/deformation/spaces.hpp

10.513 serif::geometry::deformation::SurfaceEmbedding Class Reference

Own a linear conversion from radial displacement lengths to Cartesian surface displacements.

#include <surface_embedding.hpp>

Collaboration diagram for serif::geometry::deformation::SurfaceEmbedding:



Public Types

- using InputSpace = spaces::ShapeParameters
- using OutputSpace = spaces::SurfaceDisplacement

Public Member Functions

- std::size_t input_size () const noexcept

  Nodal coordinates are independent lengths; harmonic coordinates retain their physical normalization.
- std::size_t output_size () const noexcept

  Surface vectors interleave the three Cartesian components at each node.
- std::size_t surface_node_count () const noexcept

  Expose the ordered surface-node count for independent metric and backend calculations.
- bool has_nodal_parameters () const noexcept

  An empty harmonic basis represents the identity between parameters and nodal radial lengths.
- std::span< const std::array< double, 3 > > radial_directions () const noexcept

  Reference directions determine radial motion and remain immutable when coefficients change.
- std::span< const double > reference_radii () const noexcept

  Reference radii retain physical lengths independently of the logical radius used by volume extensions.
- std::span< const std::array< double, 3 > > reference_positions () const noexcept

Preserve the exact ordered reference points so boundary metrics can reject an unrelated trace.

- std::span< const SphericalHarmonicMode > modes () const noexcept

  Harmonic coordinates use increasing degree, then increasing signed azimuthal order.

- std::span< const double > nodal_basis () const noexcept

  Return the physical row-major harmonic basis; nodal parameterizations have an implicit identity basis.

- DeformationStatus apply (std::span< const double > parameters, std::span< double > surface_↩ displacement) const

  Apply radial motion without allocation. Input and output storage must not overlap; rejected output must be discarded.

- DeformationStatus apply_transpose (std::span< const double > surface_force, std::span< double > parameter_force) const

  Transfer Cartesian surface forces to shape coordinates while preserving virtual work, without allocation or metric weighting.

Static Public Attributes

- static constexpr bool is_linear = true

Private Member Functions

- SurfaceEmbedding ()=default

  Factories validate all reference geometry before publishing an embedding.

Private Attributes

- std::vector< std::array< double, 3 > > radial_directions_
- std::vector< double > reference_radii_
- std::vector< std::array< double, 3 > > reference_positions_
- std::vector< SphericalHarmonicMode > modes_
- std::vector< double > nodal_basis_

Friends

- std::expected< SurfaceEmbedding, DeformationFailure > build_surface_embedding (const NodalRadialSurface &model, std::span< const std::array< double, 3 > > positions)

  Compile fixed radial directions from ordered physical reference positions without retaining backend objects.

- std::expected< SurfaceEmbedding, DeformationFailure > build_surface_embedding (const SphericalHarmonicSurface &model, std::span< const std::array< double, 3 > > positions)

  Sample physically normalized real harmonics and reject unresolved modes without renormalizing or dropping coordinates.

10.513.1 Detailed Description

Own a linear conversion from radial displacement lengths to Cartesian surface displacements.

Note

Directions and harmonics remain fixed on the reference surface. Finite-element interpolation of the resulting nodal vectors determines the between-node geometry.

10.513.2 Member Typedef Documentation

10.513.2.1 InputSpace

using serif::geometry::deformation::SurfaceEmbedding::InputSpace = spaces::ShapeParameters

10.513.2.2 OutputSpace

using serif::geometry::deformation::SurfaceEmbedding::OutputSpace = spaces::SurfaceDisplacement

10.513.3 Constructor & Destructor Documentation

10.513.3.1 SurfaceEmbedding()

serif::geometry::deformation::SurfaceEmbedding::SurfaceEmbedding () [private], [default]

Factories validate all reference geometry before publishing an embedding.

10.513.4 Member Function Documentation

10.513.4.1 apply()

DeformationStatus serif::geometry::deformation::SurfaceEmbedding::apply (
std::span< const double > parameters,
std::span< double > surface_displacement) const [nodiscard]

Apply radial motion without allocation. Input and output storage must not overlap; rejected output must be discarded.

10.513.4.2 apply_transpose()

DeformationStatus serif::geometry::deformation::SurfaceEmbedding::apply_transpose (
std::span< const double > surface_force,
std::span< double > parameter_force) const [nodiscard]

Transfer Cartesian surface forces to shape coordinates while preserving virtual work, without allocation or metric weighting.

10.513.4.3 has_nodal_parameters()

bool serif::geometry::deformation::SurfaceEmbedding::has_nodal_parameters () const [nodiscard], [noexcept]

An empty harmonic basis represents the identity between parameters and nodal radial lengths.

10.513.4.4 input_size()

std::size_t serif::geometry::deformation::SurfaceEmbedding::input_size () const [nodiscard], [noexcept]

Nodal coordinates are independent lengths; harmonic coordinates retain their physical normalization.

10.513.4.5 modes()

std::span< const SphericalHarmonicMode > serif::geometry::deformation::SurfaceEmbedding::modes () const [nodiscard], [noexcept]

Harmonic coordinates use increasing degree, then increasing signed azimuthal order.

10.513.4.6 nodal_basis()

std::span< const double > serif::geometry::deformation::SurfaceEmbedding::nodal_basis () const [nodiscard], [noexcept]

Return the physical row-major harmonic basis; nodal parameterizations have an implicit identity basis.

10.513.4.7 output_size()

std::size_t serif::geometry::deformation::SurfaceEmbedding::output_size () const [nodiscard], [noexcept]

Surface vectors interleave the three Cartesian components at each node.

10.513.4.8 radial_directions()

std::span< const std::array< double, 3 > > serif::geometry::deformation::SurfaceEmbedding::radial_directions () const [nodiscard], [noexcept]

Reference directions determine radial motion and remain immutable when coefficients change.

10.513.4.9 reference_positions()

std::span< const std::array< double, 3 > > serif::geometry::deformation::SurfaceEmbedding::reference_positions () const [nodiscard], [noexcept]

Preserve the exact ordered reference points so boundary metrics can reject an unrelated trace.

10.513.4.10 reference_radii()

std::span< const double > serif::geometry::deformation::SurfaceEmbedding::reference_radii () const [nodiscard], [noexcept]

Reference radii retain physical lengths independently of the logical radius used by volume extensions.

10.513.4.11 surface_node_count()

std::size_t serif::geometry::deformation::SurfaceEmbedding::surface_node_count () const [nodiscard], [noexcept]

Expose the ordered surface-node count for independent metric and backend calculations.

10.513.5 Friends And Related Symbol Documentation

10.513.5.1 build_surface_embedding [1/2]

std::expected< SurfaceEmbedding, DeformationFailure > build_surface_embedding (
const NodalRadialSurface & model,
std::span< const std::array< double, 3 > > positions) [friend]

Compile fixed radial directions from ordered physical reference positions without retaining backend objects.

10.513.5.2 build_surface_embedding [2/2]

std::expected< SurfaceEmbedding, DeformationFailure > build_surface_embedding (
const SphericalHarmonicSurface & model,
std::span< const std::array< double, 3 > > positions) [friend]

Sample physically normalized real harmonics and reject unresolved modes without renormalizing or dropping coordinates.

10.513.6 Member Data Documentation

10.513.6.1 is_linear

bool serif::geometry::deformation::SurfaceEmbedding::is_linear = true [static], [constexpr]

10.513.6.2 modes_

std::vector<SphericalHarmonicMode> serif::geometry::deformation::SurfaceEmbedding::modes_ [private]

10.513.6.3 nodal_basis_

std::vector<double> serif::geometry::deformation::SurfaceEmbedding::nodal_basis_ [private]

10.513.6.4 radial_directions_

std::vector<std::array<double, 3> > serif::geometry::deformation::SurfaceEmbedding::radial_directions_ [private]

10.513.6.5 reference_positions_

std::vector<std::array<double, 3> > serif::geometry::deformation::SurfaceEmbedding::reference_positions_ [private]

10.513.6.6 reference_radii_

std::vector<double> serif::geometry::deformation::SurfaceEmbedding::reference_radii_ [private]

The documentation for this class was generated from the following files:

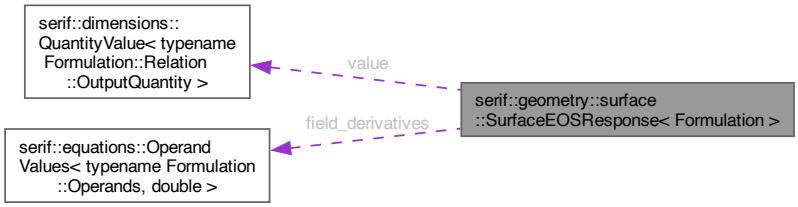
- src/include/serif/geometry/deformation/surface_embedding.hpp
- src/lib/serif/geometry/deformation/surface_embedding.cpp

10.514 serif::geometry::surface::SurfaceEOSResponse< Formulation > Struct Template Reference

An EOS value and all of its local derivatives, evaluated together at one state.

#include <linearization.hpp>

Collaboration diagram for serif::geometry::surface::SurfaceEOSResponse< Formulation >:



Public Attributes

- dimensions::QuantityValue< typename Formulation::Relation::OutputQuantity > value {0.0}
- equations::OperandValues< typename Formulation::Operands, double > field_derivatives {}

10.514.1 Detailed Description

template<PressureSurfaceFormulationType Formulation>
struct serif::geometry::surface::SurfaceEOSResponse< Formulation >

An EOS value and all of its local derivatives, evaluated together at one state.

Note

Derivatives sharing a term are accumulated; the later residual calculation supplies the sign and carrier identity.

10.514.2 Member Data Documentation

10.514.2.1 field_derivatives

template<PressureSurfaceFormulationType Formulation>
equations::OperandValues<typename Formulation::Operands, double> serif::geometry::surface::SurfaceEOSResponse< Formulation >::field_derivatives {}

10.514.2.2 value

template<PressureSurfaceFormulationType Formulation>
dimensions::QuantityValue<typename Formulation::Relation::OutputQuantity> serif::geometry::surface::SurfaceEOSResponse< Formulation >::value {0.0}

The documentation for this struct was generated from the following file:

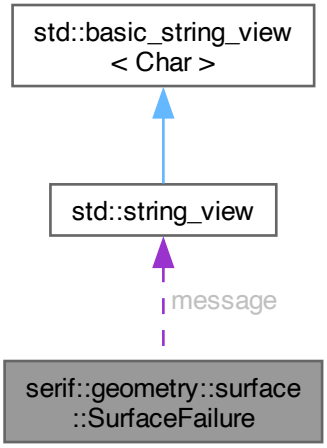
- src/include/serif/geometry/surface/linearization.hpp

10.515 serif::geometry::surface::SurfaceFailure Struct Reference

A rejected thermodynamic candidate, with a sample index when evaluating a boundary batch.

#include <linearization.hpp>

Collaboration diagram for serif::geometry::surface::SurfaceFailure:



Public Attributes

- SurfaceFailureReason reason
- std::string_view message
- std::size_t sample = std::numeric_limits<std::size_t>::max()

10.515.1 Detailed Description

A rejected thermodynamic candidate, with a sample index when evaluating a boundary batch.

10.515.2 Member Data Documentation

10.515.2.1 message

std::string_view serif::geometry::surface::SurfaceFailure::message

10.515.2.2 reason

SurfaceFailureReason serif::geometry::surface::SurfaceFailure::reason

10.515.2.3 sample

std::size_t serif::geometry::surface::SurfaceFailure::sample = std::numeric_limits<std::size_t>::max()

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/linearization.hpp

10.516 serif::geometry::deformation::keys::SurfaceMap Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

10.517 serif::geometry::deformation::keys::SurfaceMetric Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

10.518 serif::geometry::deformation::keys::SurfaceModel Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

10.519 serif::geometry::surface::detail::SurfaceOperands< Terms > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/detail/relations.hpp

10.520 serif::geometry::surface::detail::SurfaceOperands< utils::dependencies::KeyList< Terms... > > Struct Template Reference

#include <relations.hpp>

Public Types

- using Type = utils::types::TypeList<discretization::forms::Operand<Terms>...>

10.520.1 Member Typedef Documentation

10.520.1.1 Type

template<typename... Terms>
using serif::geometry::surface::detail::SurfaceOperands< utils::dependencies::KeyList< Terms... > >::Type = utils::types::TypeList<discretization::f

The documentation for this struct was generated from the following file:

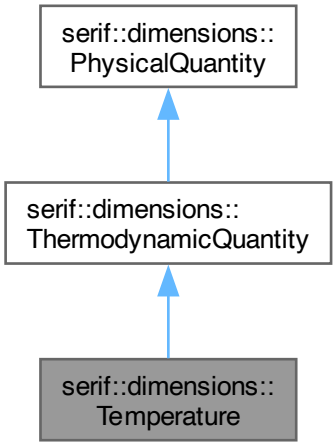
- src/include/serif/geometry/surface/detail/relations.hpp

10.521 serif::dimensions::Temperature Struct Reference

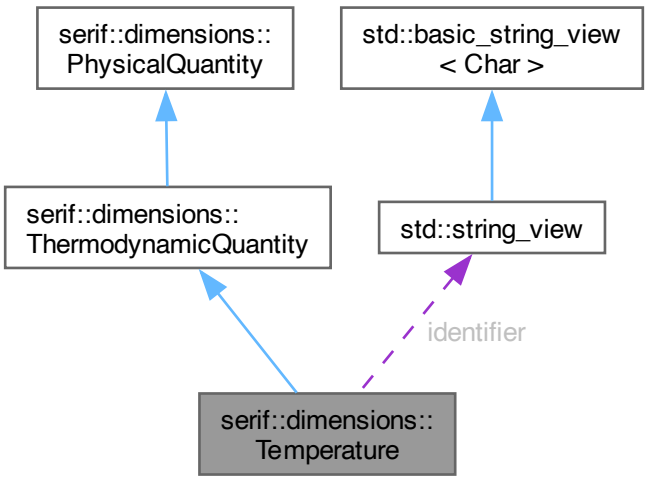
Dimension of Temperature.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Temperature:



Collaboration diagram for serif::dimensions::Temperature:



Static Public Attributes

- static constexpr std::string_view identifier = ”temperature”

10.521.1 Detailed Description

Dimension of Temperature.

10.521.2 Member Data Documentation

10.521.2.1 identifier

std::string_view serif::dimensions::Temperature::identifier = ”temperature” [static], [constexpr]

The documentation for this struct was generated from the following file:

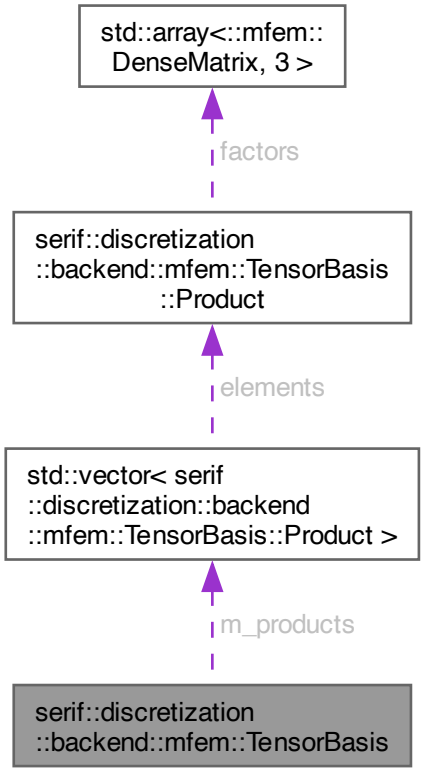
- src/include/serif/dimensions/dimensions.hpp

10.522 serif::discretization::backend::mfem::TensorBasis Class Reference

A class that represents a tensor basis for a finite element. This class is used to evaluate the basis functions and their derivatives at a set of reference points.

#include <tensor_basis.hpp>

Collaboration diagram for serif::discretization::backend::mfem::TensorBasis:



Classes

- struct Product

  A struct that represents a product of one-dimensional basis functions.

Public Member Functions

- int points () const noexcept

  Get the number of quadrature points in the Cartesian rule.
- int dofs () const noexcept

  Get the number of degrees of freedom in the tensor basis.
- int components () const noexcept

  Get the number of canonical output components per point in the tensor basis.
- void apply (const ::mfem::Vector &input, ::mfem::Vector &output, TensorWorkspace &workspace, const bool transpose=false) const

  Evaluate canonical samples, or their coordinate transpose, with caller-owned scratch.

Static Public Member Functions

- static std::shared_ptr< const TensorBasis > make (const ::mfem::FiniteElement &element, const ↩ ::mfem::IntegrationRule &rule, const ReferenceOperation operation, const int copies)

  Return no tensor plan unless its factorization agrees with MFEM's canonical basis.

Private Member Functions

- bool agrees_with (const ::mfem::FiniteElement &element, const ::mfem::IntegrationRule &rule, const ReferenceOperation operation) const

  Verify signed ordering and every requested basis entry before enabling the fast path.

Static Private Member Functions

- static void contract_axis (const ::mfem::DenseMatrix &factor, const int axis, const std::array< int, 3 > &shape, const ::mfem::Vector &input, ::mfem::Vector &output, const bool transpose)

Private Attributes

- int m_points {0}

  The number of quadrature points in the Cartesian rule.
- int m_dofs {0}

  The number of degrees of freedom in the tensor basis.
- int m_components {0}

  The number of canonical output components per point in the tensor basis.
- std::vector< Product > m_products

  The products of one-dimensional basis functions for each axis. (TOOD: consider if this should be a heap allocated vector of it this could be put on the stack somehow)

10.522.1 Detailed Description

A class that represents a tensor basis for a finite element. This class is used to evaluate the basis functions and their derivatives at a set of reference points.

10.522.2 Member Function Documentation

10.522.2.1 agrees_with()

bool serif::discretization::backend::mfem::TensorBasis::agrees_with (
const ::mfem::FiniteElement & element,
const ::mfem::IntegrationRule & rule,
const ReferenceOperation operation) const [inline], [nodiscard], [private]

Verify signed ordering and every requested basis entry before enabling the fast path.

Verify that the signed ordering and every requested basis entry agrees with MFEM's canonical basis.

Parameters

element	The finite element to verify against.
rule	The integration rule to use for evaluating the basis functions.
operation	The reference operation to apply (e.g., value, divergence).

Returns

True if the tensor basis agrees with MFEM's canonical basis, false otherwise.

10.522.2.2 apply()

```
void serif::discretization::backend::mfem::TensorBasis::apply (
            const ::mfem::Vector & input,
            ::mfem::Vector & output,
            TensorWorkspace & workspace,
            const bool transpose = false) const    [inline]
```

Evaluate canonical samples, or their coordinate transpose, with caller-owned scratch.

Apply the tensor basis to an input vector and store the results in an output vector. This function performs the tensor contraction of the input vector with the tensor basis.

Parameters

input	The input vector to be contracted with the tensor basis.
output	The output vector to store the result of the contraction. (Owned by the caller)
workspace	The workspace to use for the contraction. (Owned by the caller)
transpose	Whether to transpose the contraction (default is false).

Precondition

The input vector must have a size equal to the number of degrees of freedom in the tensor basis if transpose is false, or equal to the number of points times the number of components if transpose is true.

The output vector must be constructed

Postcondition

The output vector will have a size equal to the number of points times the number of components if transpose is false, or equal to the number of degrees of freedom in the tensor basis if transpose is true.

The output vector will contain the result of the tensor contraction of the input vector with the tensor basis.

10.522.2.3 components()

int serif::discretization::backend::mfem::TensorBasis::components () const [inline], [nodiscard], [noexcept]

Get the number of canonical output components per point in the tensor basis.

Returns

The number of canonical output components per point in the tensor basis.

10.522.2.4 contract_axis()

static void serif::discretization::backend::mfem::TensorBasis::contract_axis (
const ::mfem::DenseMatrix & factor,
const int axis,
const std::array< int, 3 > & shape,
const ::mfem::Vector & input,
::mfem::Vector & output,
const bool transpose) [inline], [static], [private]

@breif apply one factor along one tensor axis, teh other two indices will remain unchanged

Parameters

factor	The one-dimensional basis function to apply along the specified axis.
axis	The axis along which to apply the factor (0, 1, or 2).
shape	The shape of the input tensor (number of points along each axis).
input	The input vector to be contracted with the factor.
output	The output vector to store the result of the contraction.
transpose	Whether to transpose the factor matrix before applying it.

Precondition

The input vector must have a size equal to the product of the dimensions specified in the shape array.

Postcondition

The output vector will have a size equal to the product of the dimensions specified in the next array.

The output vector will contain the result of the contraction of the input vector with the factor along the specified axis.

10.522.2.5 dofs()

int serif::discretization::backend::mfem::TensorBasis::dofs () const [inline], [nodiscard], [noexcept]

Get the number of degrees of freedom in the tensor basis.

Returns

The number of degrees of freedom in the tensor basis.

10.522.2.6 make()

static std::shared_ptr< const TensorBasis > serif::discretization::backend::mfem::TensorBasis::make (
const ::mfem::FiniteElement & element,
const ::mfem::IntegrationRule & rule,
const ReferenceOperation operation,
const int copies) [inline], [static], [nodiscard]

Return no tensor plan unless its factorization agrees with MFEM's canonical basis.

Make a tensor basis for a given finite element, integration rule, reference operation, and number of copies. This function returns a shared pointer to a TensorBasis object that can be used to do efficient evaluation of the basis functions and their derivatives at reference points.

Parameters

element	The finite element for which to construct the tensor basis.
rule	The integration rule to use for evaluating the basis functions.
operation	The reference operation to apply (e.g., value, divergence).
copies	The number of copies of the basis functions.

Returns

A shared pointer to a TensorBasis object if the tensor basis can be constructed, or an empty shared pointer if it cannot.

10.522.2.7 points()

int serif::discretization::backend::mfem::TensorBasis::points () const [inline], [nodiscard], [noexcept]

Get the number of quadrature points in the Cartesian rule.

Returns

The number of quadrature points in the Cartesian rule.

10.522.3 Member Data Documentation

10.522.3.1 m_components

int serif::discretization::backend::mfem::TensorBasis::m_components {0} [private]

The number of canonical output components per point in the tensor basis.

10.522.3.2 m_dofs

int serif::discretization::backend::mfem::TensorBasis::m_dofs {0} [private]

The number of degrees of freedom in the tensor basis.

10.522.3.3 m_points

int serif::discretization::backend::mfem::TensorBasis::m_points {0} [private]

The number of quadrature points in the Cartesian rule.

10.522.3.4 m_products

std::vector<Product> serif::discretization::backend::mfem::TensorBasis::m_products [private]

The products of one-dimensional basis functions for each axis. (TOOD: consider if this should be a heap allocated vector of it this could be put on the stack somehow)

The documentation for this class was generated from the following file:

- src/include/serif/discretization/backend/mfem/tensor_basis.hpp

10.523 serif::discretization::backend::mfem::TensorBasisCache Class Reference

Share immutable tensor plans within one operator; copied rules are keyed by their contents.

#include <tensor_basis.hpp>

Collaboration diagram for serif::discretization::backend::mfem::TensorBasisCache:



Classes

- struct Entry

  A struct that represents an entry in the TensorBasisCache. Each entry contains a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

Public Member Functions

- std::shared_ptr< const TensorBasis > get (const ::mfem::FiniteElement &element, const ::mfem::↩ IntegrationRule &rule, ReferenceOperation operation, const int copies)

  Retrieve a verified plan, caching unsupported combinations as well.
- std::size_t size () const noexcept

  Get the number of entries in the TensorBasisCache.
- void clear ()

  Clear the TensorBasisCache, removing all entries.

Private Attributes

- std::vector< Entry > m_entries

  The entries in the TensorBasisCache, each containing a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

10.523.1 Detailed Description

Share immutable tensor plans within one operator; copied rules are keyed by their contents.

A cache for TensorBasis objects to avoid repeated construction of the same tensor basis. This class stores a vector of "Entry" structs which each contain a non-owning pointer to some element, options, a shared pointer to a TensorBasis, and the points used to construct the TensorBasis.

10.523.2 Member Function Documentation

10.523.2.1 clear()

void serif::discretization::backend::mfem::TensorBasisCache::clear () [inline]

Clear the TensorBasisCache, removing all entries.

10.523.2.2 get()

std::shared_ptr< const TensorBasis > serif::discretization::backend::mfem::TensorBasisCache::get (
const ::mfem::FiniteElement & element,
const ::mfem::IntegrationRule & rule,
ReferenceOperation operation,
const int copies) [inline], [nodiscard]

Retrieve a verified plan, caching unsupported combinations as well.

Retrieve a verified tensor basis, caching unsupported combinations as well.

Parameters

element	The finite element for which to retrieve the tensor basis.
rule	The integration rule to use for evaluating the basis functions.
operation	The reference operation to apply (e.g., value, divergence).
copies	The number of copies of the basis functions.

Returns

A shared pointer to a TensorBasis object if the tensor basis can be constructed, or an empty shared pointer if it cannot.

Todo I think it would make sense, at some point, to switch this to an std::optional return

10.523.2.3 size()

std::size_t serif::discretization::backend::mfem::TensorBasisCache::size () const [inline], [nodiscard], [noexcept]

Get the number of entries in the TensorBasisCache.

Returns

The number of entries in the cache.

10.523.3 Member Data Documentation

10.523.3.1 m_entries

std::vector<Entry> serif::discretization::backend::mfem::TensorBasisCache::m_entries [private]

The entries in the TensorBasisCache, each containing a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

The documentation for this class was generated from the following file:

- src/include/serif/discretization/backend/mfem/tensor_basis.hpp

10.524 serif::operators::backend::mfem::TensorBasisCache Class Reference

Share immutable tensor plans within one operator; copied rules are keyed by their contents.

#include <tensor_basis.hpp>

Collaboration diagram for serif::operators::backend::mfem::TensorBasisCache:



Classes

- struct Entry

  A struct that represents an entry in the TensorBasisCache. Each entry contains a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

Public Member Functions

- std::shared_ptr< const TensorBasis > get (const ::mfem::FiniteElement &element, const ::mfem::↩ IntegrationRule &rule, ReferenceOperation operation, const int copies)

  Retrieve a verified plan, caching unsupported combinations as well.
- std::size_t size () const noexcept

  Get the number of entries in the TensorBasisCache.
- void clear ()

  Clear the TensorBasisCache, removing all entries.

Private Attributes

- std::vector< Entry > m_entries

  The entries in the TensorBasisCache, each containing a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

10.524.1 Detailed Description

Share immutable tensor plans within one operator; copied rules are keyed by their contents.

A cache for TensorBasis objects to avoid repeated construction of the same tensor basis. This class stores a vector of "Entry" structs which each contain a non-owning pointer to some element, options, a shared pointer to a TensorBasis, and the points used to construct the TensorBasis.

10.524.2 Member Function Documentation

10.524.2.1 clear()

void serif::discretization::backend::mfem::TensorBasisCache::clear () [inline]

Clear the TensorBasisCache, removing all entries.

10.524.2.2 get()

std::shared_ptr< const TensorBasis > serif::discretization::backend::mfem::TensorBasisCache::get (
const ::mfem::FiniteElement & element,
const ::mfem::IntegrationRule & rule,
ReferenceOperation operation,
const int copies) [inline], [nodiscard]

Retrieve a verified plan, caching unsupported combinations as well.

Retrieve a verified tensor basis, caching unsupported combinations as well.

Parameters

element	The finite element for which to retrieve the tensor basis.
rule	The integration rule to use for evaluating the basis functions.
operation	The reference operation to apply (e.g., value, divergence).
copies	The number of copies of the basis functions.

Returns

A shared pointer to a TensorBasis object if the tensor basis can be constructed, or an empty shared pointer if it cannot.

Todo I think it would make sense, at some point, to switch this to an std::optional return

10.524.2.3 size()

std::size_t serif::discretization::backend::mfem::TensorBasisCache::size () const [inline], [nodiscard], [noexcept]

Get the number of entries in the TensorBasisCache.

Returns

The number of entries in the cache.

10.524.3 Member Data Documentation

10.524.3.1 m_entries

std::vector<Entry> serif::discretization::backend::mfem::TensorBasisCache::m_entries [private]

The entries in the TensorBasisCache, each containing a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

The documentation for this class was generated from the following file:

- src/include/serif/discretization/backend/mfem/tensor_basis.hpp

10.525 serif::discretization::backend::mfem::TensorWorkspace Struct Reference

A workspace for tensor contractions to avoid repeated allocations and pack memory (TODO: does this actual pack memory in a useful manner when using mfem vectors? Probably something to look into).

#include <tensor_basis.hpp>

Public Attributes

- ::mfem::Vector coefficients

  The coefficients of the basis functions at the reference points.
- ::mfem::Vector first

  The first contraction of the coefficients with the basis functions at the reference points.
- ::mfem::Vector second

  The second contraction of the coefficients with the basis functions at the reference points.

10.525.1 Detailed Description

A workspace for tensor contractions to avoid repeated allocations and pack memory (TODO: does this actual pack memory in a useful manner when using mfem vectors? Probably something to look into).

10.525.2 Member Data Documentation

10.525.2.1 coefficients

::mfem::Vector serif::discretization::backend::mfem::TensorWorkspace::coefficients

The coefficients of the basis functions at the reference points.

10.525.2.2 first

::mfem::Vector serif::discretization::backend::mfem::TensorWorkspace::first

The first contraction of the coefficients with the basis functions at the reference points.

10.525.2.3 second

::mfem::Vector serif::discretization::backend::mfem::TensorWorkspace::second

The second contraction of the coefficients with the basis functions at the reference points.

The documentation for this struct was generated from the following file:

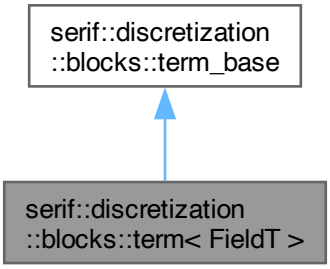
- src/include/serif/discretization/backend/mfem/tensor_basis.hpp

10.526 serif::discretization::blocks::term< FieldT > Struct Template Reference

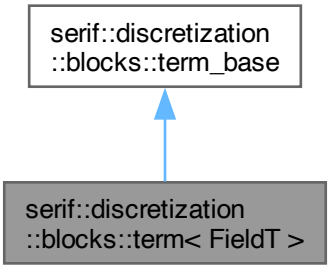
The term struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Note that generally you should always use term rather than term_base as term associates to a particular field at compile time.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::term< FieldT >:



Collaboration diagram for serif::discretization::blocks::term< FieldT >:



Public Types

- using field_type = FieldT

10.526.1 Detailed Description

template<typename FieldT>
struct serif::discretization::blocks::term< FieldT >

The term struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Note that generally you should always use term rather than term_base as term associates to a particular field at compile time.

10.526.2 Member Typedef Documentation

10.526.2.1 field_type

template<typename FieldT>
using serif::discretization::blocks::term< FieldT >::field_type = FieldT

The documentation for this struct was generated from the following file:

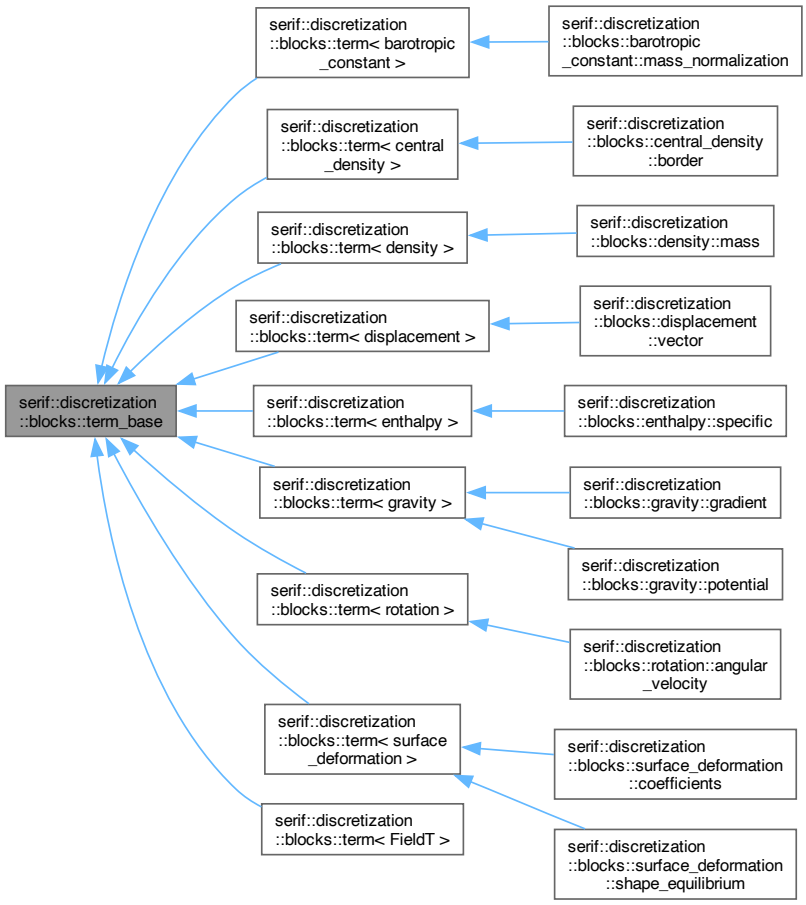
- src/include/serif/discretization/blocks/fields.hpp

10.527 serif::discretization::blocks::term_base Struct Reference

The term_base struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Generally you should always use term rather than term_base as term associates to a particular field at compile time.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::term_base:



10.527.1 Detailed Description

The term_base struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Generally you should always use term rather than term_base as term associates to a particular field at compile time.

The documentation for this struct was generated from the following file:

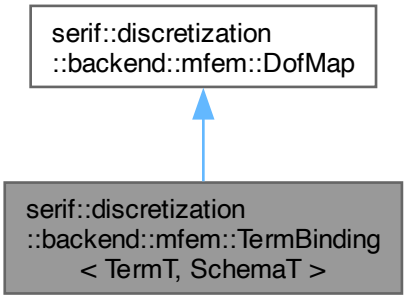
- src/include/serif/discretization/blocks/fields.hpp

10.528 serif::discretization::backend::mfem::TermBinding< TermT, SchemaT > Class Template Reference

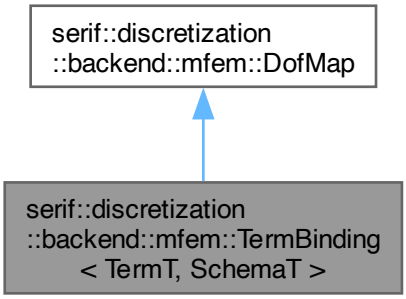
A runtime binding between a term descriptor and its MFEM finite-element space. This is used to project down the compile time term descriptors into runtime reinterpretations which can then be used to extract things like the list of dofs, the element support, and the mesh. Note that this is a child of DofMap, therefor all term bindings are also dof maps.

#include <term_binding.hpp>

Inheritance diagram for serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >:



Collaboration diagram for serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >:



Public Types

- using Term = TermT
- using Schema = SchemaT

Public Member Functions

- TermBinding (::mfem::FiniteElementSpace &space)

  Construct a term binding from an MFEM finite-element space. This requires that the term be a valid spatial term (SpatialTerm)
- TermBinding (::mfem::Mesh &mesh)

  Construct a term binding from an MFEM mesh. This requires that the term be a valid global scalar term (GlobalScalarTerm). E.g. the only way you can construct such a term binding is for things like the total mass, angular momentum, etc...

Public Member Functions inherited from serif::discretization::backend::mfem::DofMap

- DofMap (const DofMap &)=delete
- DofMap & operator= (const DofMap &)=delete
- DofMap (DofMap &&)=delete
- DofMap & operator= (DofMap &&)=delete
- ::mfem::FiniteElementSpace & space () const

  Return the borrowed finite-element space associated with this DOF map.
- ::mfem::Mesh & mesh () const noexcept

  Return the borrowed mesh associated with this DOF map.
- int full_size () const noexcept

  Return the number of full-space DOFs, including those not supported by the element support.
- int size () const noexcept

  Return the number of supported DOFs, which is the size of the reduced coefficient vector.
- std::optional< int > reduced_dof (const int full_dof) const

  Located a full space dof in the reduced space. E.g if I ask where is dof at index x in the full space this will tell me what index it is in the reduced space. If the dof is not supported this will return std::nullopt.
- bool supports (const int element) const

  Check whether an element is supported by this dof map.
- void validate () const

  Validate the dof map.
- void validate_values (const ::mfem::Vector &values) const

  Check supported coefficient count and finiteness.
- ::mfem::Vector gather (const ::mfem::Vector &full) const

  Restrict a full coefficient vector to supported DOFs.
- ::mfem::Vector scatter (const ::mfem::Vector &reduced) const

  Expand supported coefficients with zero outside their support.
- void element_values (const int element, const ::mfem::Vector &reduced, ::mfem::Vector &values) const

  Extract element coefficients, including primal orientation transforms.
- void add_element (const int element, ::mfem::Vector &values, ::mfem::Vector &full) const

  Apply dual orientation transforms and add an element residual.

Private Member Functions

- static ::mfem::Array< int > element_support (const ::mfem::Mesh &mesh)

  Translate the mesh into an element support array for the particular term binding.

Static Private Member Functions

- ::mfem::Mesh & checked_mesh (const ::mfem::FiniteElementSpace &space)

  Check that the mesh associated with a finite-element space is valid for term binding construction.

Additional Inherited Members

Protected Member Functions inherited from serif::discretization::backend::mfem::DofMap

- DofMap (::mfem::FiniteElementSpace &space, const ::mfem::Array< int > &element_support)
- DofMap (::mfem::Mesh &mesh, const ::mfem::Array< int > &element_support)
- ~DofMap ()=default

10.528.1 Detailed Description

template<typename TermT, domain::schema::IsSchema SchemaT>
requires (blocks::SpatialTerm<TermT> || blocks::GlobalScalarTerm<TermT>)
class serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >

A runtime binding between a term descriptor and its MFEM finite-element space. This is used to project down the compile time term descriptors into runtime reinterpretations which can then be used to extract things like the list of dofs, the element support, and the mesh. Note that this is a child of DofMap, therefor all term bindings are also dof maps.

Template Parameters

TermT	The term descriptor type to bind to. Note that if the FiniteElementSpace is not a global scalar then this must be a spatial term descriptor (SpatialTerm)
SchemaT	The domain schema type to bind to. This must be a valid schema (IsSchema)

10.528.2 Member Typedef Documentation

10.528.2.1 Schema

template<typename TermT, domain::schema::IsSchema SchemaT>
using serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >::Schema = SchemaT

10.528.2.2 Term

template<typename TermT, domain::schema::IsSchema SchemaT>
using serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >::Term = TermT

10.528.3 Constructor & Destructor Documentation

10.528.3.1 TermBinding() [1/2]

template<typename TermT, domain::schema::IsSchema SchemaT>
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >::TermBinding (
::mfem::FiniteElementSpace & space) [inline], [explicit]

Construct a term binding from an MFEM finite-element space. This requires that the term be a valid spatial term (SpatialTerm)

Parameters

space	The finite-element space to bind to.

10.528.3.2 TermBinding() [2/2]

template<typename TermT, domain::schema::IsSchema SchemaT>
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >::TermBinding (
::mfem::Mesh & mesh) [inline], [explicit]

Construct a term binding from an MFEM mesh. This requires that the term be a valid global scalar term (GlobalScalarTerm). E.g. the only way you can construct such a term binding is for things like the total mass, angular momentum, etc...

Parameters

mesh	The mesh to bind to, note that the mesh must be three-dimensional and have a space dimension of three.

10.528.4 Member Function Documentation

10.528.4.1 checked_mesh()

template<typename TermT, domain::schema::IsSchema SchemaT>
::mfem::Mesh & serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >::checked_mesh (
const ::mfem::FiniteElementSpace & space) [inline], [static], [private]

Check that the mesh associated with a finite-element space is valid for term binding construction.

Parameters

space	The finite-element space to check.

Returns

The mesh associated with the finite-element space.

10.528.4.2 element_support()

template<typename TermT, domain::schema::IsSchema SchemaT>
static ::mfem::Array< int > serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >::element_support (
const ::mfem::Mesh & mesh) [inline], [private]

Translate the mesh into an element support array for the particular term binding.

Parameters

mesh	the mesh to translate into an element support array

Returns

An array where each entry is 1 if the corresponding element is supported by the term binding and 0 otherwise. This array is parallel to the mesh elements ordering array.

The documentation for this class was generated from the following file:

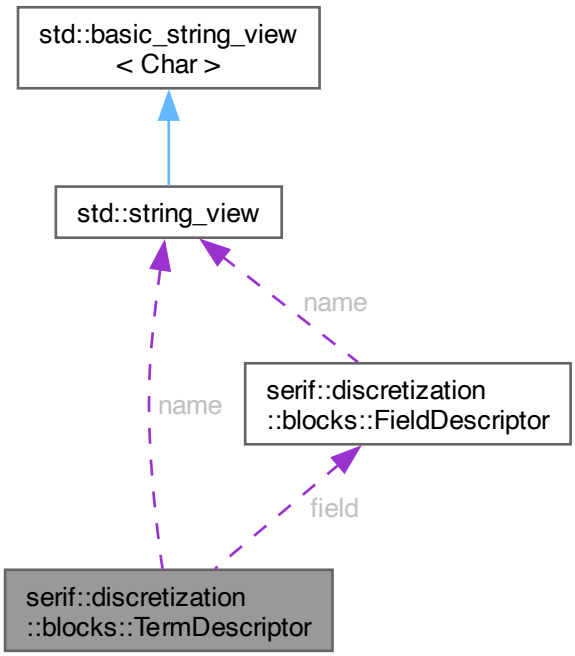
- src/include/serif/discretization/backend/mfem/term_binding.hpp

10.529 serif::discretization::blocks::TermDescriptor Struct Reference

Runtime descriptor for a term, containing its associated field and name. Two TermDescriptors are equal if they have identical fields and names.

#include <runtime.hpp>

Collaboration diagram for serif::discretization::blocks::TermDescriptor:



Public Attributes

- FieldDescriptor field

  Each term belongs to a field, and this descriptor identifies that field.
- std::string_view name

  The name of the term, used for runtime identification.

Friends

- constexpr bool operator== (const TermDescriptor &, const TermDescriptor &) noexcept=default

  Compare two TermDescriptors for equality based on their fields and names.

10.529.1 Detailed Description

Runtime descriptor for a term, containing its associated field and name. Two TermDescriptors are equal if they have identical fields and names.

10.529.2 Friends And Related Symbol Documentation

10.529.2.1 operator==

```
bool operator== (
            const TermDescriptor & ,
            const TermDescriptor & )    [friend]
```

Compare two TermDescriptors for equality based on their fields and names.

Parameters

lhs	The left-hand side TermDescriptor.
rhs	The right-hand side TermDescriptor.

Returns

True if both the fields and names of the TermDescriptors are equal, false otherwise.

10.529.3 Member Data Documentation

10.529.3.1 field

FieldDescriptor serif::discretization::blocks::TermDescriptor::field

Each term belongs to a field, and this descriptor identifies that field.

10.529.3.2 name

std::string_view serif::discretization::blocks::TermDescriptor::name

The name of the term, used for runtime identification.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/runtime.hpp

10.530 serif::equations::TestOperandTraits< T, typename > Struct Template Reference

#include <contributions.hpp>

Public Types

- using type = discretization::forms::Operand<typename T::Test>

10.530.1 Member Typedef Documentation

10.530.1.1 type

template<typename T, typename = void>
using serif::equations::TestOperandTraits< T, typename >::type = discretization::forms::Operand<typename T::Test>

The documentation for this struct was generated from the following file:

- src/include/serif/equations/contributions.hpp

10.531 serif::equations::TestOperandTraits< T, std::void_t< typename T::TestOperand > > Struct Template Reference

#include <contributions.hpp>

Public Types

- using type = typename T::TestOperand
- using type

10.531.1 Member Typedef Documentation

10.531.1.1 type [1/2]

using serif::equations::TestOperandTraits< T, typename >::type

10.531.1.2 type [2/2]

template<typename T>
using serif::equations::TestOperandTraits< T, std::void_t< typename T::TestOperand > >::type = typename T::TestOperand

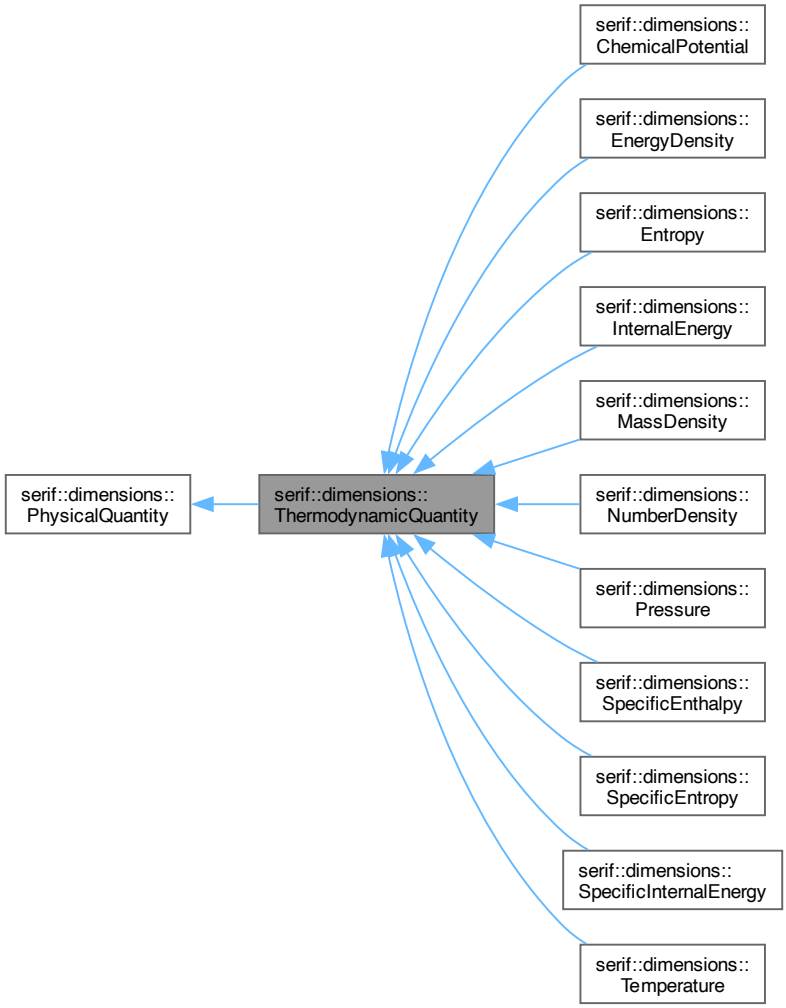
The documentation for this struct was generated from the following file:

- src/include/serif/equations/contributions.hpp

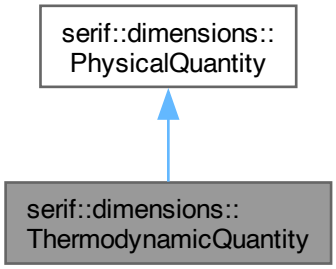
10.532 serif::dimensions::ThermodynamicQuantity Struct Reference

#include <quantities.hpp>

Inheritance diagram for serif::dimensions::ThermodynamicQuantity:



Collaboration diagram for serif::dimensions::ThermodynamicQuantity:



10.532.1 Detailed Description

@breif Slightly more specific type for thermodynamic types. Note both of these are empty at runtime, these are only useful as compile time tools to control what code the compiler generates.

The documentation for this struct was generated from the following file:

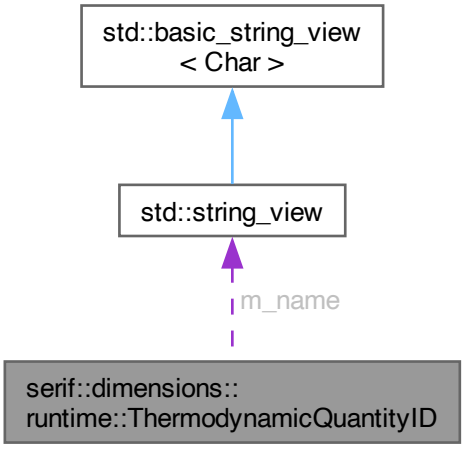
- src/include/serif/dimensions/quantities.hpp

10.533 serif::dimensions::runtime::ThermodynamicQuantityID Class Reference

Runtime resolvable ID for a thermodynamic quantity. Used in conjunction with concepts to resolve compile time thermodynamic types to runtime values.

#include <runtime.hpp>

Collaboration diagram for serif::dimensions::runtime::ThermodynamicQuantityID:



Public Member Functions

- constexpr ThermodynamicQuantityID (const std::string_view name) noexcept
- constexpr std::string_view name () const noexcept

Private Attributes

- std::string_view m_name

Friends

- constexpr bool operator== (const ThermodynamicQuantityID &, const ThermodynamicQuantityID &) noexcept=default

10.533.1 Detailed Description

Runtime resolvable ID for a thermodynamic quantity. Used in conjunction with concepts to resolve compile time thermodynamic types to runtime values.

10.533.2 Constructor & Destructor Documentation

10.533.2.1 ThermodynamicQuantityID()

serif::dimensions::runtime::ThermodynamicQuantityID::ThermodynamicQuantityID (
const std::string_view name) [inline], [explicit], [constexpr], [noexcept]

10.533.3 Member Function Documentation

10.533.3.1 name()

std::string_view serif::dimensions::runtime::ThermodynamicQuantityID::name () const [inline], [nodiscard], [constexpr], [noexcept]

10.533.4 Friends And Related Symbol Documentation

10.533.4.1 operator==

bool operator== (
const ThermodynamicQuantityID & ,
const ThermodynamicQuantityID &) [friend]

10.533.5 Member Data Documentation

10.533.5.1 m_name

std::string_view serif::dimensions::runtime::ThermodynamicQuantityID::m_name [private]

The documentation for this class was generated from the following file:

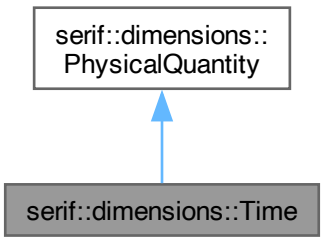
- src/include/serif/dimensions/runtime/runtime.hpp

10.534 serif::dimensions::Time Struct Reference

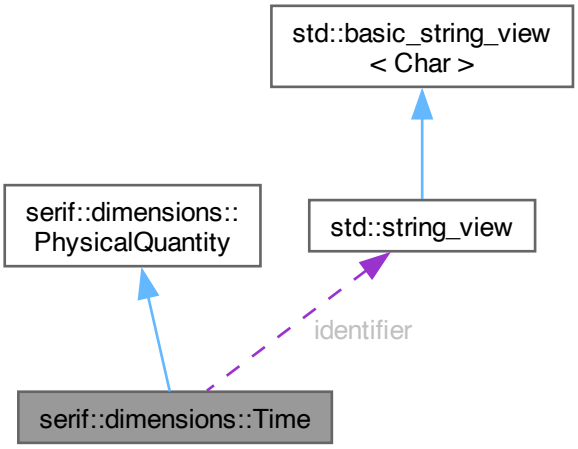
Dimension of Time.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Time:



Collaboration diagram for serif::dimensions::Time:



Static Public Attributes

- static constexpr std::string_view identifier = ”time”

10.534.1 Detailed Description

Dimension of Time.

10.534.2 Member Data Documentation

10.534.2.1 identifier

std::string_view serif::dimensions::Time::identifier = ”time” [static], [constexpr]

The documentation for this struct was generated from the following file:

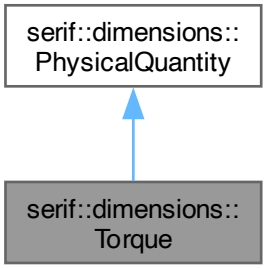
- src/include/serif/dimensions/dimensions.hpp

10.535 serif::dimensions::Torque Struct Reference

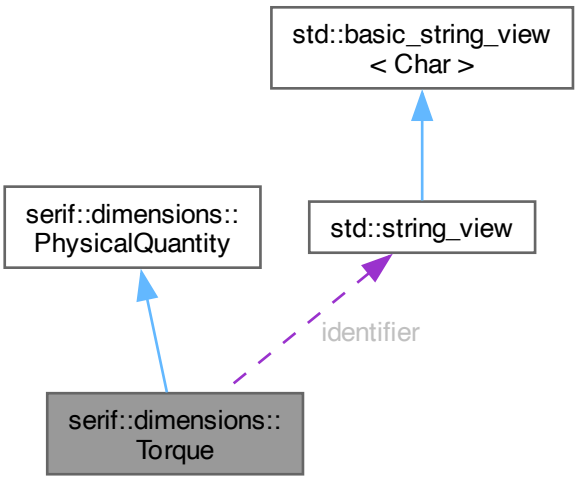
Dimension of Torque.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Torque:



Collaboration diagram for serif::dimensions::Torque:



Static Public Attributes

- static constexpr std::string_view identifier = "torque"

10.535.1 Detailed Description

Dimension of Torque.

10.535.2 Member Data Documentation

10.535.2.1 identifier

std::string_view serif::dimensions::Torque::identifier = "torque" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/dimensions.hpp

10.536 serif::utils::types::TypeList< Ts > Struct Template Reference

#include <type_list.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/utils/types/type_list.hpp

10.537 serif::geometry::surface::detail::UniqueTerms< Terms > Struct Template Reference

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/detail/relations.hpp

10.538 serif::geometry::surface::detail::UniqueTerms< utils::types::TypeList< Terms... > > Struct Template Reference

#include <relations.hpp>

Public Types

- using Type = utils::dependencies::CombinedKeyList<utils::dependencies::KeyList<Terms>...>

10.538.1 Member Typedef Documentation

10.538.1.1 Type

template<typename... Terms>
using serif::geometry::surface::detail::UniqueTerms< utils::types::TypeList< Terms... > >::Type = utils::dependencies::CombinedKeyList<utils::dep

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/surface/detail/relations.hpp

10.539 serif::utils::dependencies::detail::UsedResource< GraphT > Struct Template Reference

#include <cache_traits.hpp>

Classes

- struct Predicate

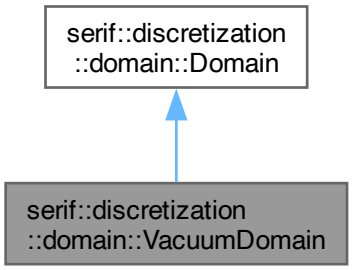
The documentation for this struct was generated from the following file:

- src/include/serif/utils/dependencies/detail/cache_traits.hpp

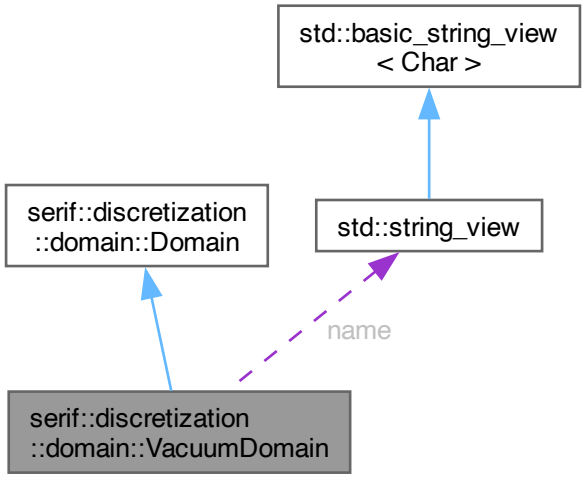
10.540 serif::discretization::domain::VacuumDomain Struct Reference

#include <types.hpp>

Inheritance diagram for serif::discretization::domain::VacuumDomain:



Collaboration diagram for serif::discretization::domain::VacuumDomain:



Static Public Attributes

- static constexpr std::string_view name = "vacuum"

10.540.1 Member Data Documentation

10.540.1.1 name

std::string_view serif::discretization::domain::VacuumDomain::name = "vacuum" [static], [constexpr]

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/domain/types.hpp

10.541 serif::geometry::deformation::keys::VacuumModel Struct Reference

#include <computations.hpp>

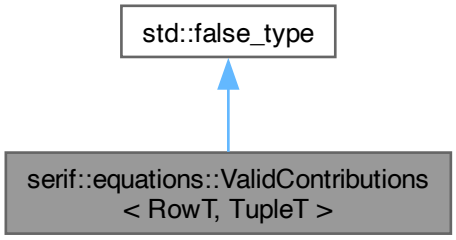
The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

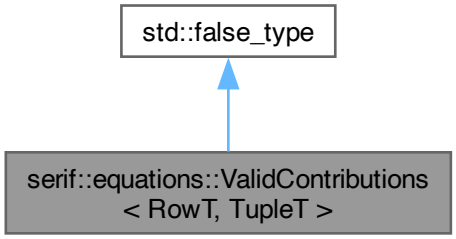
10.542 serif::equations::ValidContributions< RowT, TupleT > Struct Template Reference

#include <concepts.hpp>

Inheritance diagram for serif::equations::ValidContributions< RowT, TupleT >:



Collaboration diagram for serif::equations::ValidContributions< RowT, TupleT >:



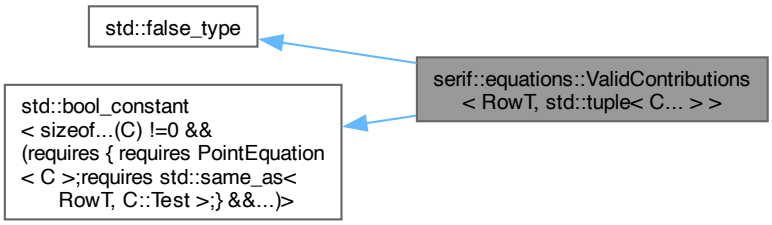
The documentation for this struct was generated from the following file:

- src/include/serif/equations/concepts.hpp

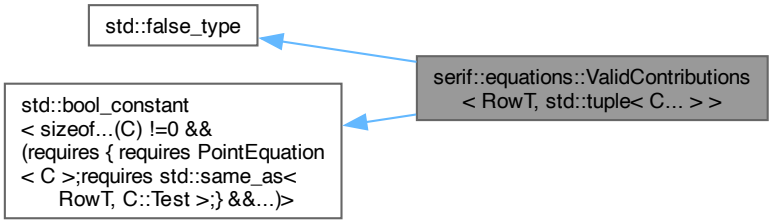
10.543 serif::equations::ValidContributions< RowT, std::tuple< C... > > Struct Template Reference

#include <concepts.hpp>

Inheritance diagram for serif::equations::ValidContributions< RowT, std::tuple< C... > >:



Collaboration diagram for serif::equations::ValidContributions< RowT, std::tuple< C... > >:



The documentation for this struct was generated from the following file:

- src/include/serif/equations/concepts.hpp

10.544 serif::discretization::forms::Operation::Value Struct Reference

Operation tag for evaluating a value.

#include <operands.hpp>

10.544.1 Detailed Description

Operation tag for evaluating a value.

The documentation for this struct was generated from the following file:

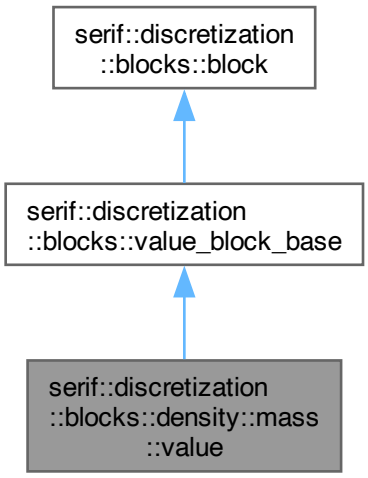
- src/include/serif/discretization/forms/operands.hpp

10.545 serif::discretization::blocks::density::mass::value Struct Reference

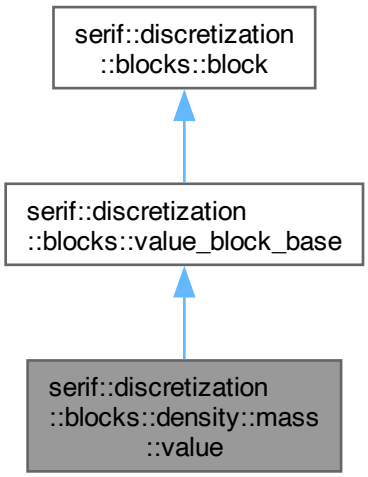
The value block is a block that contains the value of the mass density term. It is a scalar field that is associated with the mass density of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::density::mass::value:



Collaboration diagram for serif::discretization::blocks::density::mass::value:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::value_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.545.1 Detailed Description

The value block is a block that contains the value of the mass density term. It is a scalar field that is associated with the mass density of the system.

The documentation for this struct was generated from the following file:

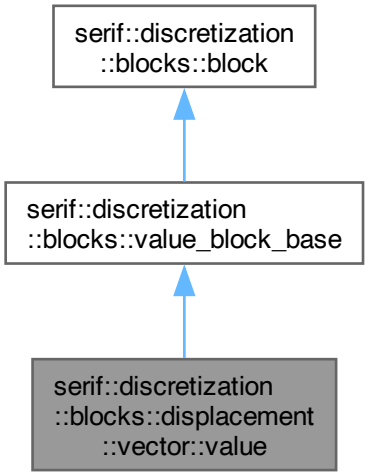
- src/include/serif/discretization/blocks/fields.hpp

10.546 serif::discretization::blocks::displacement::vector::value Struct Reference

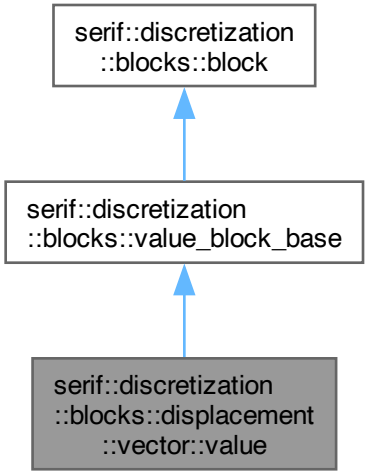
The value block is a block that contains the value of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::displacement::vector::value:



Collaboration diagram for serif::discretization::blocks::displacement::vector::value:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::value_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.546.1 Detailed Description

The value block is a block that contains the value of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces.

The documentation for this struct was generated from the following file:

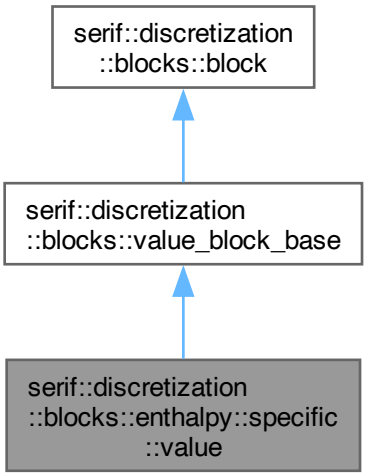
- src/include/serif/discretization/blocks/fields.hpp

10.547 serif::discretization::blocks::enthalpy::specific::value Struct Reference

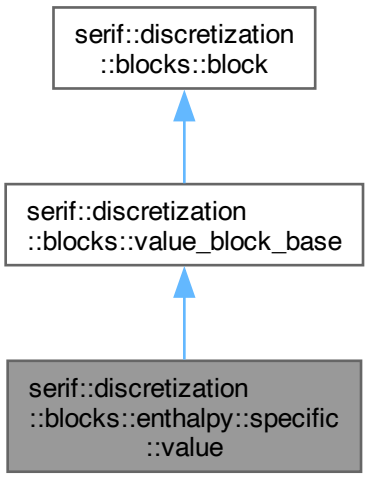
The value block is a block that contains the value of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::enthalpy::specific::value:



Collaboration diagram for serif::discretization::blocks::enthalpy::specific::value:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::value_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.547.1 Detailed Description

The value block is a block that contains the value of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system.

The documentation for this struct was generated from the following file:

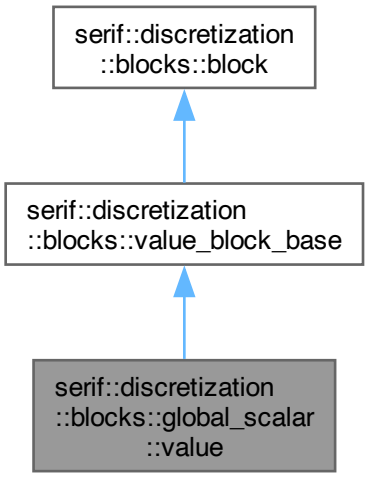
- src/include/serif/discretization/blocks/fields.hpp

10.548 serif::discretization::blocks::global_scalar::value Struct Reference

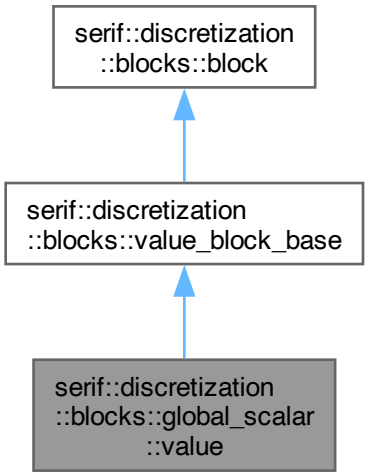
The value block is a block that contains the value of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the actual value of the mass of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::global_scalar::value:



Collaboration diagram for serif::discretization::blocks::global_scalar::value:



Static Public Attributes

- static constexpr int static_block_size = 1

Because this is a global scalar unknown we can be sure at compile time that the block size is 1 and we can use this to optimize the linear algebra operations on this block.

Static Public Attributes inherited from serif::discretization::blocks::value_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.548.1 Detailed Description

The value block is a block that contains the value of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the actual value of the mass of the system.

10.548.2 Member Data Documentation

10.548.2.1 static_block_size

int serif::discretization::blocks::global_scalar::value::static_block_size = 1 [static], [constexpr]

Because this is a global scalar unknown we can be sure at compile time that the block size is 1 and we can use this to optimize the linear algebra operations on this block.

The documentation for this struct was generated from the following file:

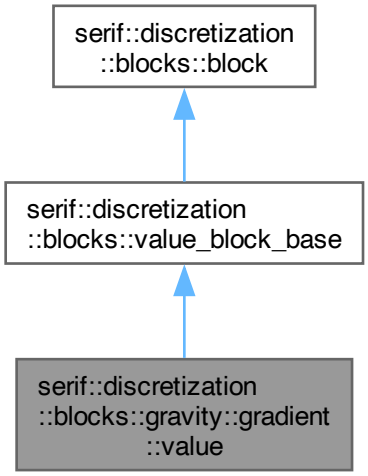
- src/include/serif/discretization/blocks/fields.hpp

10.549 serif::discretization::blocks::gravity::gradient::value Struct Reference

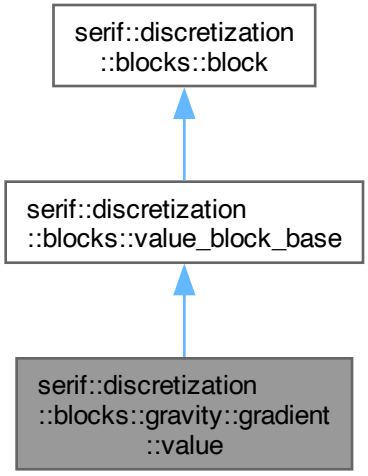
The value block is a block that contains the value of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::gravity::gradient::value:



Collaboration diagram for serif::discretization::blocks::gravity::gradient::value:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::value_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.549.1 Detailed Description

The value block is a block that contains the value of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system.

The documentation for this struct was generated from the following file:

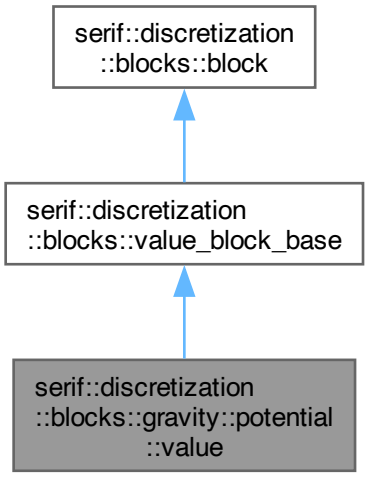
- src/include/serif/discretization/blocks/fields.hpp

10.550 serif::discretization::blocks::gravity::potential::value Struct Reference

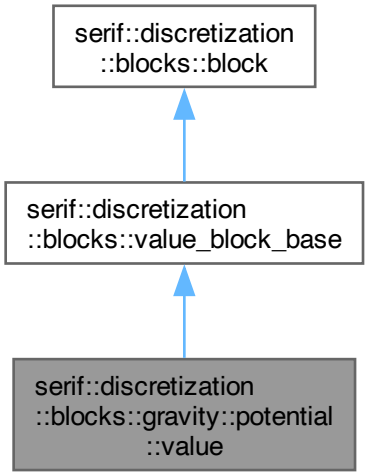
The value block is a block that contains the value of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::gravity::potential::value:



Collaboration diagram for serif::discretization::blocks::gravity::potential::value:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::value_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.550.1 Detailed Description

The value block is a block that contains the value of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system.

The documentation for this struct was generated from the following file:

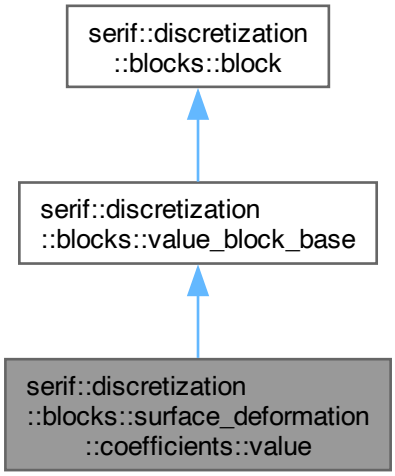
- src/include/serif/discretization/blocks/fields.hpp

10.551 serif::discretization::blocks::surface_deformation::coefficients::value Struct Reference

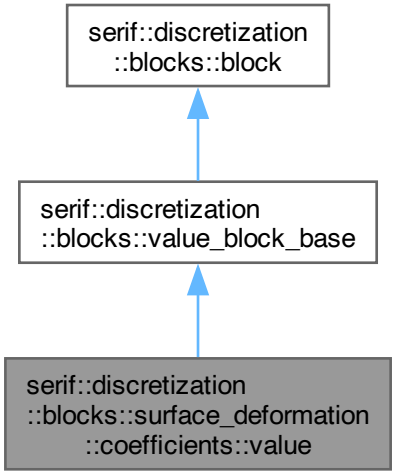
The actual values of the coefficients for whatever surface representation is being used.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::surface_deformation::coefficients::value:



Collaboration diagram for serif::discretization::blocks::surface_deformation::coefficients::value:



Additional Inherited Members

Static Public Attributes inherited from serif::discretization::blocks::value_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.551.1 Detailed Description

The actual values of the coefficients for whatever surface representation is being used.

The documentation for this struct was generated from the following file:

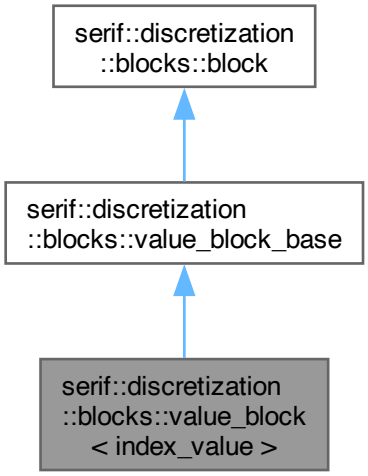
- src/include/serif/discretization/blocks/fields.hpp

10.552 serif::discretization::blocks::value_block< index_value > Struct Template Reference

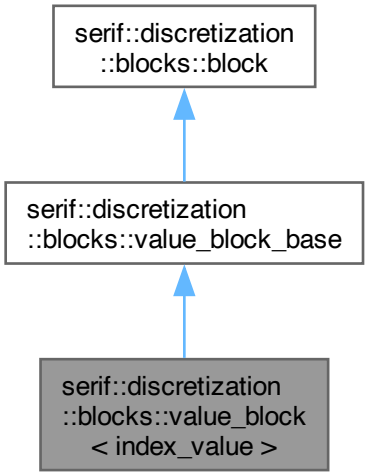
Final class for value blocks with a fixed index, derived from value_block_base.

#include <base.hpp>

Inheritance diagram for serif::discretization::blocks::value_block< index_value >:



Collaboration diagram for serif::discretization::blocks::value_block< index_value >:



Public Member Functions

- constexpr operator int () const noexcept

Static Public Attributes

- static constexpr int index = index_value

Static Public Attributes inherited from serif::discretization::blocks::value_block_base

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.552.1 Detailed Description

template<int index_value>
struct serif::discretization::blocks::value_block< index_value >

Final class for value blocks with a fixed index, derived from value_block_base.

10.552.2 Member Function Documentation

10.552.2.1 operator int()

template<int index_value>
serif::discretization::blocks::value_block< index_value >::operator int () const [inline], [constexpr], [noexcept]

10.552.3 Member Data Documentation

10.552.3.1 index

template<int index_value>
int serif::discretization::blocks::value_block< index_value >::index = index_value [static], [constexpr]

The documentation for this struct was generated from the following file:

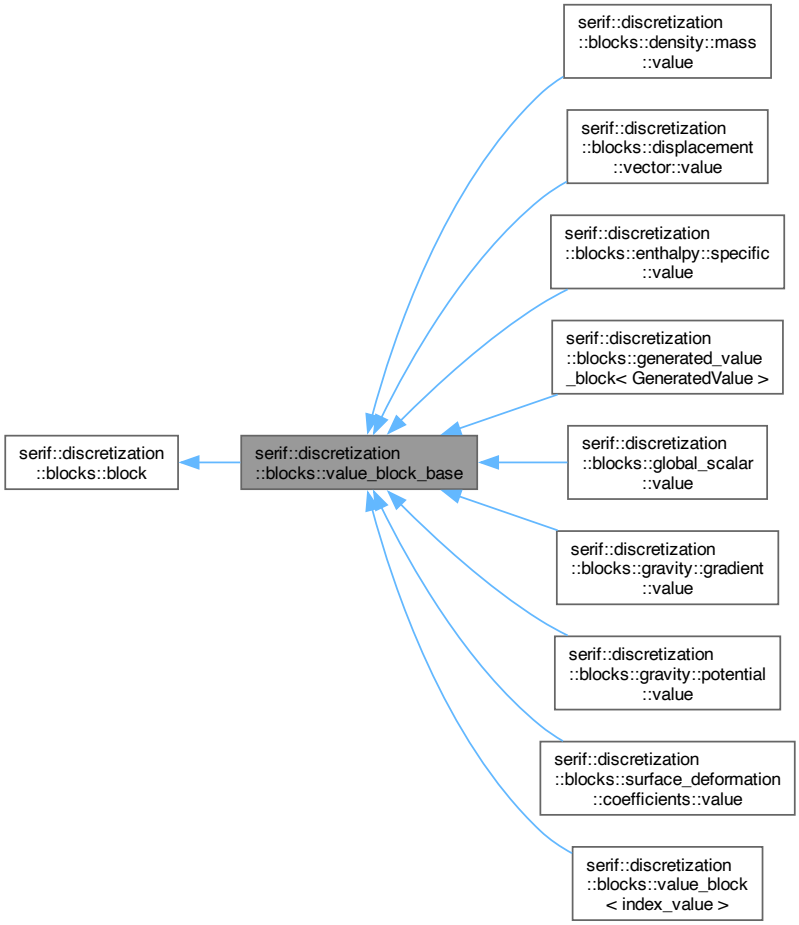
- src/include/serif/discretization/blocks/base.hpp

10.553 serif::discretization::blocks::value_block_base Struct Reference

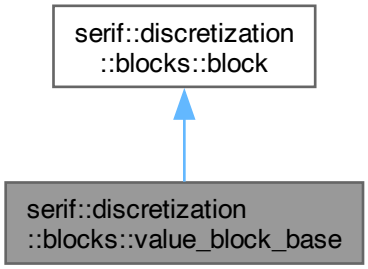
Base class for value blocks, derived from block.

#include <base.hpp>

Inheritance diagram for serif::discretization::blocks::value_block_base:



Collaboration diagram for serif::discretization::blocks::value_block_base:



Static Public Attributes

- static constexpr int static_block_size = DYNAMIC_BLOCK_SIZE

10.553.1 Detailed Description

Base class for value blocks, derived from block.

10.553.2 Member Data Documentation

10.553.2.1 static_block_size

int serif::discretization::blocks::value_block_base::static_block_size = DYNAMIC_BLOCK_SIZE [static], [constexpr]

The documentation for this struct was generated from the following file:

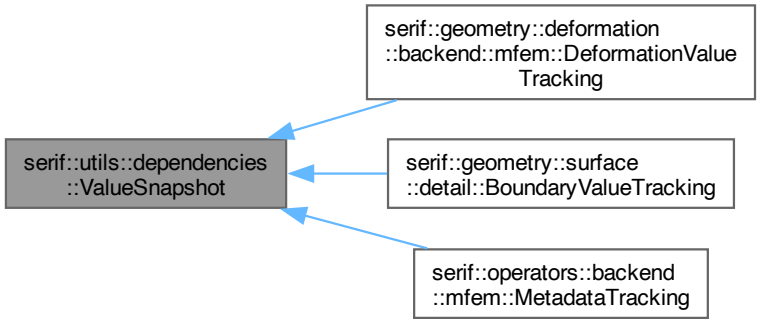
- src/include/serif/discretization/blocks/base.hpp

10.554 serif::utils::dependencies::ValueSnapshot Struct Reference

Default tracking owns an exact value copy; noncopyable resources supply another policy.

#include <snapshot.hpp>

Inheritance diagram for serif::utils::dependencies::ValueSnapshot:



Public Member Functions

- template<typename T>
  requires std::copy_constructible<T> && std::equality_comparable<T>
  T capture (const T &value) const

  Keep the accepted value independent of subsequent edits to a borrowed source.
- template<typename T>
  requires std::equality_comparable<T>
  bool matches (const T &captured, const T &value) const

  Compare content rather than a hash that could silently miss a change.

10.554.1 Detailed Description

Default tracking owns an exact value copy; noncopyable resources supply another policy.

10.554.2 Member Function Documentation

10.554.2.1 capture()

template<typename T>
requires std::copy_constructible<T> && std::equality_comparable<T>
T serif::utils::dependencies::ValueSnapshot::capture (
const T & value) const [inline], [nodiscard]

Keep the accepted value independent of subsequent edits to a borrowed source.

10.554.2.2 matches()

```
template<typename T>
requires std::equality_comparable<T>
bool serif::utils::dependencies::ValueSnapshot::matches (
                const T & captured,
                const T & value) const    [inline], [nodiscard]
```

Compare content rather than a hash that could silently miss a change.

The documentation for this struct was generated from the following file:

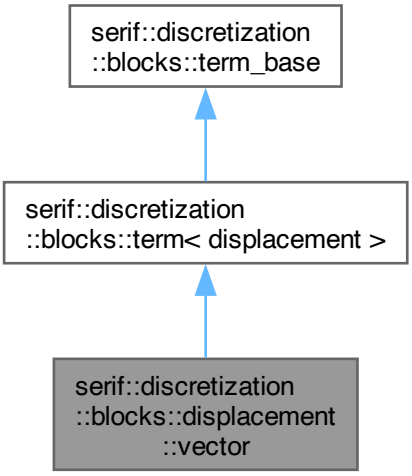
- src/include/serif/utils/dependencies/snapshot.hpp

10.555 serif::discretization::blocks::displacement::vector Struct Reference

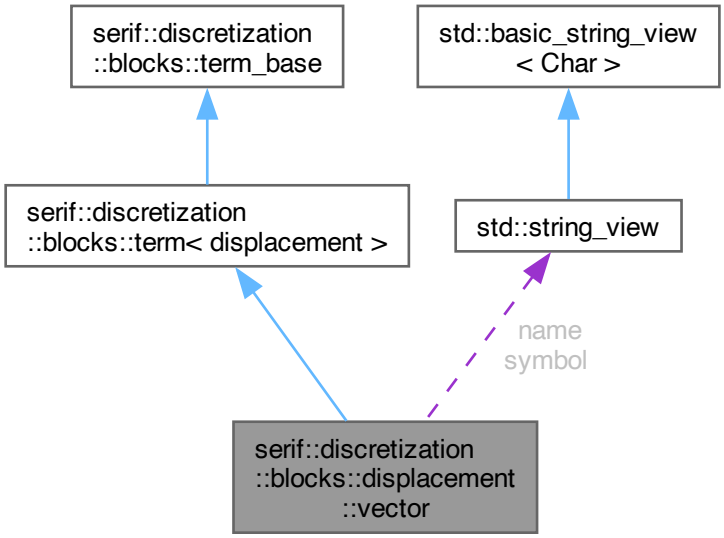
the displacement vector term is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain. This is a vector field that is associated with the mapping of the domain from the reference to physical spaces. E.g. if there is some domain O_ref which n nodal positions x^i_ref then there is some other domain O_phys with n nodel positions such that x^i_phys = x^i_ref + d^i where d^i is the displacement vector at node i.

#include <fields.hpp>

Inheritance diagram for serif::discretization::blocks::displacement::vector:



Collaboration diagram for serif::discretization::blocks::displacement::vector:



Classes

- struct residual

  The residual block is a block that contains the residual of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces.
- struct value

  The value block is a block that contains the value of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces.

Public Types

- using FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::H1, 3, 1>

  3 vector H1 space (C0 continuous). Read the template parameters as follows: 1) H1 space, 2) 3 vector, 3) rank 1 tensor field
- using Support = domain::AllDomains

  We support displacement over the entire domain since we want to be able to map the reference domain to the physical domain over the entire domain. This enables both kelvin compactification but also the smooth transition from the stellar domain to the vacuum domain.
- using Quantity = dimensions::Length

  We represent displacement as a length quantity.

Public Types inherited from serif::discretization::blocks::term< displacement >

- using field_type

Static Public Attributes

- static constexpr std::string_view name = meta::type_name<vector>()

  Compile time reflected name of the term.
- static constexpr std::string_view symbol = "d"

  Compile time symbol used for the term.

10.555.1 Detailed Description

the displacement vector term is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain. This is a vector field that is associated with the mapping of the domain from the reference to physical spaces. E.g. if there is some domain O_ref which n nodal positions x^i_ref then there is some other domain O_phys with n nodel positions such that x^i_phys = x^i_ref + d^i where d^i is the displacement vector at node i.

Note

Generally a particular model will not actually model its geometry with the full displacement field as an unknown. Rather, the displacement system will generally report some surface representation and prescription (say Isobaric with a spherical harmonic expansion) to the solver. So there may be say 25 spherical harmonic coefficients for the solver to actually solver for. Those are then analytically mapped to the full displacement field using some interior and exterior extension. For more information see the surface and deformation systems.

10.555.2 Member Typedef Documentation

10.555.2.1 FunctionalSpace

using serif::discretization::blocks::displacement::vector::FunctionalSpace = functional_space::DiscreteFunctionalSpace<functional_space::H1, 3, 1>

3 vector H1 space (C0 continuous). Read the template parameters as follows: 1) H1 space, 2) 3 vector, 3) rank 1 tensor field

10.555.2.2 Quantity

using serif::discretization::blocks::displacement::vector::Quantity = dimensions::Length

We represent displacement as a length quantity.

10.555.2.3 Support

using serif::discretization::blocks::displacement::vector::Support = domain::AllDomains

We support displacement over the entire domain since we want to be able to map the reference domain to the physical domain over the entire domain. This enables both kelvin compactification but also the smooth transition from the stellar domain to the vacuum domain.

10.555.3 Member Data Documentation

10.555.3.1 name

std::string_view serif::discretization::blocks::displacement::vector::name = meta::type_name<vector>() [static], [constexpr]

Compile time reflected name of the term.

10.555.3.2 symbol

std::string_view serif::discretization::blocks::displacement::vector::symbol = "d" [static], [constexpr]

Compile time symbol used for the term.

The documentation for this struct was generated from the following file:

- src/include/serif/discretization/blocks/fields.hpp

10.556 serif::utils::types::VectorBase< T > Struct Template Reference

#include <tensor.hpp>

Public Types

- using Type = T

10.556.1 Member Typedef Documentation

10.556.1.1 Type

template<typename T>
using serif::utils::types::VectorBase< T >::Type = T

The documentation for this struct was generated from the following file:

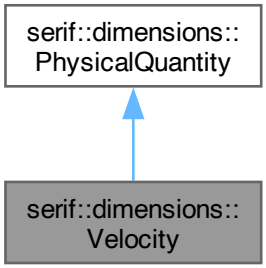
- src/include/serif/utils/types/tensor/tensor.hpp

10.557 serif::dimensions::Velocity Struct Reference

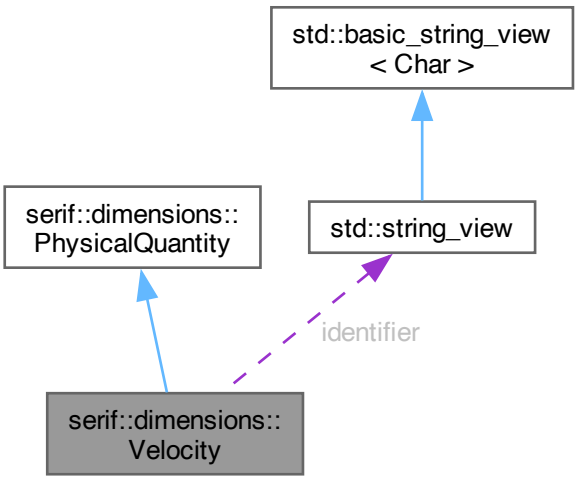
Dimension of Velocity.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Velocity:



Collaboration diagram for serif::dimensions::Velocity:



Static Public Attributes

- static constexpr std::string_view identifier = "velocity"

10.557.1 Detailed Description

Dimension of Velocity.

10.557.2 Member Data Documentation

10.557.2.1 identifier

std::string_view serif::dimensions::Velocity::identifier = "velocity" [static], [constexpr]

The documentation for this struct was generated from the following file:

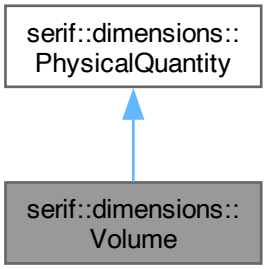
- src/include/serif/dimensions/dimensions.hpp

10.558 serif::dimensions::Volume Struct Reference

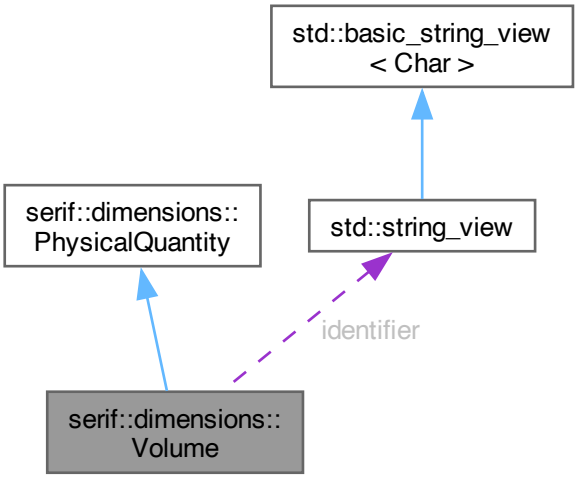
Dimension of Volume.

#include <dimensions.hpp>

Inheritance diagram for serif::dimensions::Volume:



Collaboration diagram for serif::dimensions::Volume:



Static Public Attributes

- static constexpr std::string_view identifier = ”volume”

10.558.1 Detailed Description

Dimension of Volume.

10.558.2 Member Data Documentation

10.558.2.1 identifier

std::string_view serif::dimensions::Volume::identifier = ”volume” [static], [constexpr]

The documentation for this struct was generated from the following file:

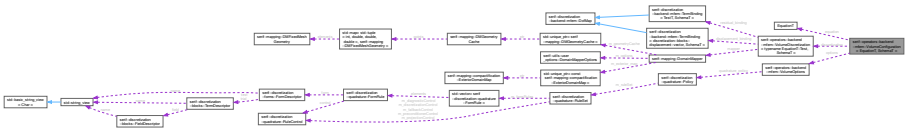
- src/include/serif/dimensions/dimensions.hpp

10.559 serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT > Struct Template Reference

Stable equation ownership also retains temporary and move-only contribution trees.

#include <volume_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >:



Public Types

- using DisplacementBinding = typename VolumeDiscretization<typename EquationT::Test, SchemaT>↩ ::DisplacementBinding

Public Attributes

- EquationT equation
- VolumeDiscretization< typename EquationT::Test, SchemaT > discretization
- VolumeOptions options

10.559.1 Detailed Description

template<VolumeEquation EquationT, typename SchemaT>
struct serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >

Stable equation ownership also retains temporary and move-only contribution trees.

10.559.2 Member Typedef Documentation

10.559.2.1 DisplacementBinding

template<VolumeEquation EquationT, typename SchemaT>
using serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >::DisplacementBinding = typename VolumeDiscretization<typename EquationT::Test, SchemaT>::DisplacementBinding

10.559.3 Member Data Documentation

10.559.3.1 discretization

template<VolumeEquation EquationT, typename SchemaT>
VolumeDiscretization<typename EquationT::Test, SchemaT> serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >::discretization

10.559.3.2 equation

template<VolumeEquation EquationT, typename SchemaT>
EquationT serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >::equation

10.559.3.3 options

template<VolumeEquation EquationT, typename SchemaT>
VolumeOptions serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >::options

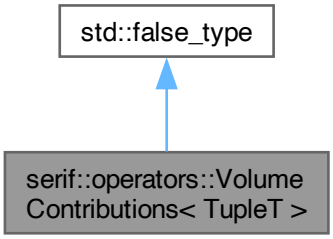
The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

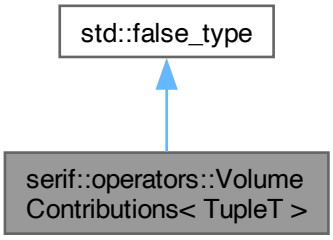
10.560 serif::operators::VolumeContributions< TupleT > Struct Template Reference

#include <concepts.hpp>

Inheritance diagram for serif::operators::VolumeContributions< TupleT >:



Collaboration diagram for serif::operators::VolumeContributions< TupleT >:



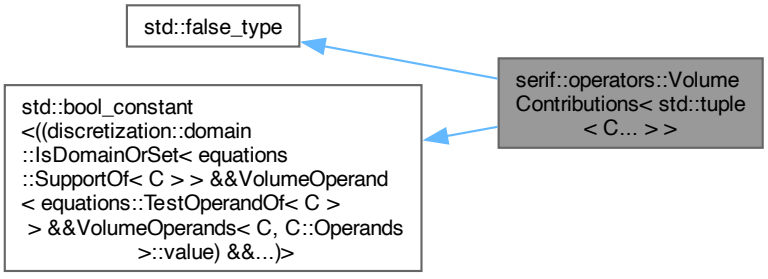
The documentation for this struct was generated from the following file:

- src/include/serif/operators/concepts.hpp

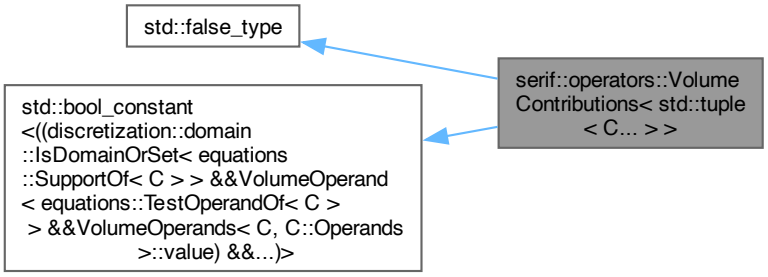
10.561 serif::operators::VolumeContributions< std::tuple< C... > > Struct Template Reference

#include <concepts.hpp>

Inheritance diagram for serif::operators::VolumeContributions< std::tuple< C... > >:



Collaboration diagram for serif::operators::VolumeContributions< std::tuple< C... > >:



The documentation for this struct was generated from the following file:

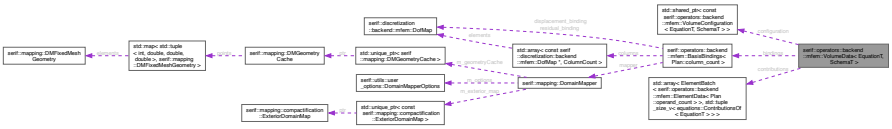
- src/include/serif/operators/concepts.hpp

10.562 serif::operators::backend::mfem::VolumeData< EquationT, SchemaT > Struct Template Reference

One accepted result contains ordered contribution batches and the assembled residual.

#include <volume_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >:



Public Types

- using Equation = EquationT
- using Schema = SchemaT
- using Plan = EquationPlan<EquationT>

Public Attributes

- std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > configuration
- BasisBindings< Plan::column_count > bindings
- std::array< ElementBatch< ElementData< Plan::operand_count > >, std::tuple_size_v< equations::ContributionsOf< EquationT > > > contributions
- ::mfem::Vector residual

10.562.1 Detailed Description

template<VolumeEquation EquationT, typename SchemaT>
struct serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >

One accepted result contains ordered contribution batches and the assembled residual.

10.562.2 Member Typedef Documentation

10.562.2.1 Equation

template<VolumeEquation EquationT, typename SchemaT>
using serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >::Equation = EquationT

10.562.2.2 Plan

template<VolumeEquation EquationT, typename SchemaT>
using serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >::Plan = EquationPlan<EquationT>

10.562.2.3 Schema

template<VolumeEquation EquationT, typename SchemaT>
using serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >::Schema = SchemaT

10.562.3 Member Data Documentation

10.562.3.1 bindings

template<VolumeEquation EquationT, typename SchemaT>
BasisBindings<Plan::column_count> serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >::bindings

10.562.3.2 configuration

template<VolumeEquation EquationT, typename SchemaT>
std::shared_ptr<const VolumeConfiguration<EquationT, SchemaT> > serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >::configuration

10.562.3.3 contributions

template<VolumeEquation EquationT, typename SchemaT>
std::array<ElementBatch<ElementData<Plan::operand_count> >, std::tuple_size_v<equations::ContributionsOf<EquationT> > > serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >::contributions

10.562.3.4 residual

template<VolumeEquation EquationT, typename SchemaT>
::mfem::Vector serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >::residual

The documentation for this struct was generated from the following file:

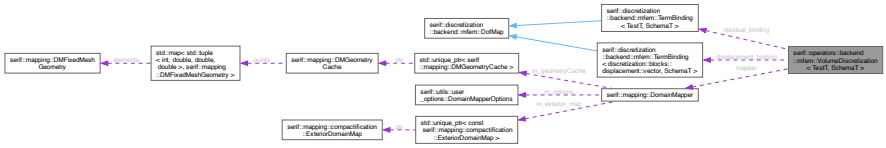
- src/include/serif/operators/backend/mfem/volume_data.hpp

10.563 serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT > Struct Template Reference

Borrowed discretization objects must outlive their graph and its accepted results.

#include <volume_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >:



Public Types

- using RowBinding = discretization::backend::mfem::TermBinding<TestT, SchemaT>
- using DisplacementBinding = discretization::backend::mfem::TermBinding<discretization::blocks::displacement::vector, SchemaT>

Public Attributes

- const RowBinding & residual_binding
- const DisplacementBinding & displacement_binding
- const ::mfem::GridFunction & compactification
- const mapping::DomainMapper & mapper
- const ::mfem::FiniteElementSpace * compactification_space {compactification.FESpace()}
- long compactification_sequence {compactification_space == nullptr ? -1 : compactification_space->GetSequence()}

10.563.1 Detailed Description

template<typename TestT, typename SchemaT>
struct serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >

Borrowed discretization objects must outlive their graph and its accepted results.

10.563.2 Member Typedef Documentation

10.563.2.1 DisplacementBinding

template<typename TestT, typename SchemaT>
using serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >::DisplacementBinding = discretization::backend::mfem::TermBind
SchemaT>

10.563.2.2 RowBinding

template<typename TestT, typename SchemaT>
using serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >::RowBinding = discretization::backend::mfem::TermBinding<Test
SchemaT>

10.563.3 Member Data Documentation

10.563.3.1 compactification

template<typename TestT, typename SchemaT>
const ::mfem::GridFunction& serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >::compactification

10.563.3.2 compactification_sequence

template<typename TestT, typename SchemaT>
long serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >::compactification_sequence {compactification_space
== nullptr ? -1 : compactification_space->GetSequence()}

10.563.3.3 compactification_space

template<typename TestT, typename SchemaT>
const ::mfem::FiniteElementSpace* serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >::compactification↩
_space {compactification.FESpace()}

10.563.3.4 displacement_binding

template<typename TestT, typename SchemaT>
const DisplacementBinding& serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >::displacement_binding

10.563.3.5 mapper

template<typename TestT, typename SchemaT>
const mapping::DomainMapper& serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >::mapper

10.563.3.6 residual_binding

template<typename TestT, typename SchemaT>
const RowBinding& serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >::residual_binding

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.564 serif::geometry::deformation::keys::VolumeDisplacement Struct Reference

#include <computations.hpp>

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

10.565 serif::geometry::deformation::spaces::VolumeDisplacement Struct Reference

Displacement coefficients in the declared volume discretization.

#include <spaces.hpp>

10.565.1 Detailed Description

Displacement coefficients in the declared volume discretization.

The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/spaces.hpp

10.566 serif::geometry::deformation::keys::VolumeExtension Struct Reference

#include <computations.hpp>

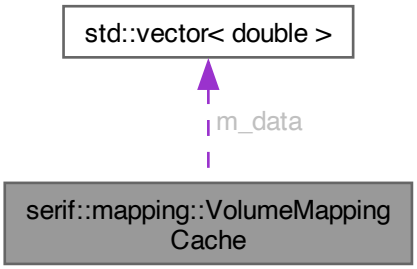
The documentation for this struct was generated from the following file:

- src/include/serif/geometry/deformation/computations.hpp

10.567 serif::mapping::VolumeMappingCache Class Reference

#include <volume_cache.hpp>

Collaboration diagram for serif::mapping::VolumeMappingCache:



Public Member Functions

- void set_size (int point_count, int dimension)
- void store (int point, VolumeMappingContext &context)
- void load (int point, VolumeMappingContext &context) const
- void load_inverse_jacobian (int point, mfem::DenseMatrix &inverse) const
- int get_point_count () const noexcept
- int get_dimension () const noexcept

Private Member Functions

- void validate_point (int point) const
- const double * get_point_data (int point) const

Private Attributes

- std::vector< double > m_data
- int m_point_count {0}
- int m_dimension {0}
- int m_point_stride {0}

10.567.1 Member Function Documentation

10.567.1.1 get_dimension()

int serif::mapping::VolumeMappingCache::get_dimension () const [nodiscard], [noexcept]

10.567.1.2 get_point_count()

int serif::mapping::VolumeMappingCache::get_point_count () const [nodiscard], [noexcept]

10.567.1.3 get_point_data()

const double * serif::mapping::VolumeMappingCache::get_point_data (
int point) const [nodiscard], [private]

10.567.1.4 load()

void serif::mapping::VolumeMappingCache::load (
int point,
VolumeMappingContext & context) const

10.567.1.5 load_inverse_jacobian()

void serif::mapping::VolumeMappingCache::load_inverse_jacobian (
int point,
mfem::DenseMatrix & inverse) const

10.567.1.6 set_size()

void serif::mapping::VolumeMappingCache::set_size (
int point_count,
int dimension)

10.567.1.7 store()

void serif::mapping::VolumeMappingCache::store (
int point,
VolumeMappingContext & context)

10.567.1.8 validate_point()

void serif::mapping::VolumeMappingCache::validate_point (
int point) const [private]

10.567.2 Member Data Documentation

10.567.2.1 m_data

std::vector<double> serif::mapping::VolumeMappingCache::m_data [private]

10.567.2.2 m_dimension

int serif::mapping::VolumeMappingCache::m_dimension {0} [private]

10.567.2.3 m_point_count

int serif::mapping::VolumeMappingCache::m_point_count {0} [private]

10.567.2.4 m_point_stride

int serif::mapping::VolumeMappingCache::m_point_stride {0} [private]

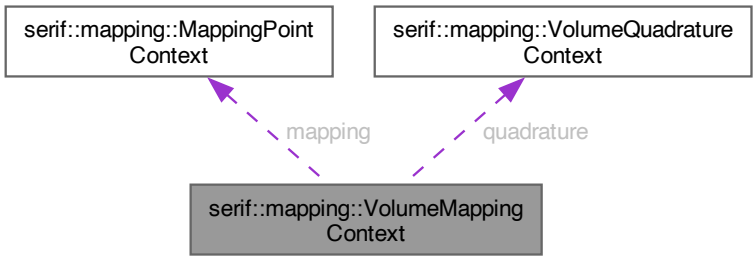
The documentation for this class was generated from the following files:

- src/include/serif/mapping/volume_cache.hpp
- src/lib/serif/mapping/volume_cache.cpp

10.568 serif::mapping::VolumeMappingContext Struct Reference

#include <context.hpp>

Collaboration diagram for serif::mapping::VolumeMappingContext:



Public Attributes

- MappingPointContext mapping
- VolumeQuadratureContext quadrature

10.568.1 Member Data Documentation

10.568.1.1 mapping

MappingPointContext serif::mapping::VolumeMappingContext::mapping

10.568.1.2 quadrature

VolumeQuadratureContext serif::mapping::VolumeMappingContext::quadrature

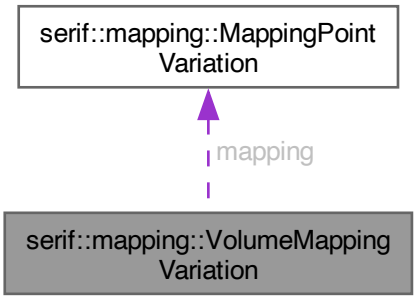
The documentation for this struct was generated from the following file:

- src/include/serif/mapping/context.hpp

10.569 serif::mapping::VolumeMappingVariation Struct Reference

#include <variation.hpp>

Collaboration diagram for serif::mapping::VolumeMappingVariation:



Public Attributes

- MappingPointVariation mapping
- mfem::DenseMatrix inverse_element_jacobian_variation
- double weight_variation {0.0}

10.569.1 Member Data Documentation

10.569.1.1 inverse_element_jacobian_variation

mfem::DenseMatrix serif::mapping::VolumeMappingVariation::inverse_element_jacobian_variation

10.569.1.2 mapping

MappingPointVariation serif::mapping::VolumeMappingVariation::mapping

10.569.1.3 weight_variation

double serif::mapping::VolumeMappingVariation::weight_variation {0.0}

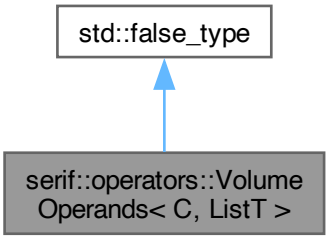
The documentation for this struct was generated from the following file:

- src/include/serif/mapping/variation.hpp

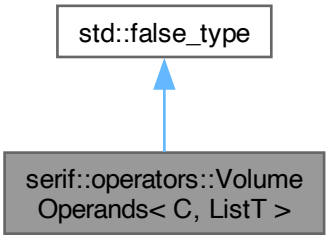
10.570 serif::operators::VolumeOperands< C, ListT > Struct Template Reference

#include <concepts.hpp>

Inheritance diagram for serif::operators::VolumeOperands< C, ListT >:



Collaboration diagram for serif::operators::VolumeOperands< C, ListT >:



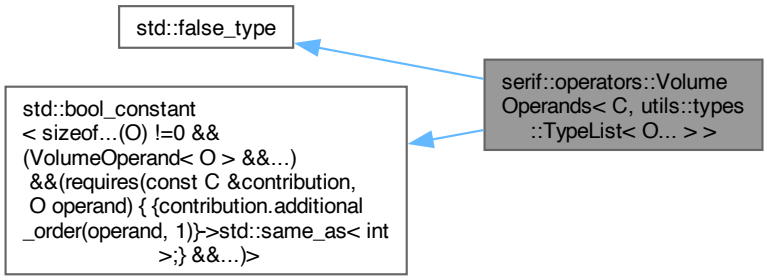
The documentation for this struct was generated from the following file:

- src/include/serif/operators/concepts.hpp

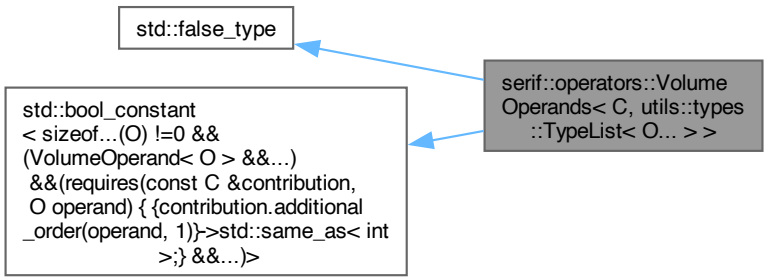
10.571 serif::operators::VolumeOperands< C, utils::types::TypeList< O... > > Struct Template Reference

#include <concepts.hpp>

Inheritance diagram for serif::operators::VolumeOperands< C, utils::types::TypeList< O... > >:



Collaboration diagram for serif::operators::VolumeOperands< C, utils::types::TypeList< O... > >:



The documentation for this struct was generated from the following file:

- src/include/serif/operators/concepts.hpp

10.572 serif::operators::backend::mfem::VolumeOptions Struct Reference

Numerical choices fixed when the graph is constructed.

#include <volume_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::VolumeOptions:



Public Attributes

- discretization::quadrature::Policy quadrature_policy {{}}
- ContractionMode contraction_mode {ContractionMode::automatic}

10.572.1 Detailed Description

Numerical choices fixed when the graph is constructed.

10.572.2 Member Data Documentation

10.572.2.1 contraction_mode

ContractionMode serif::operators::backend::mfem::VolumeOptions::contraction_mode {ContractionMode::automatic}

10.572.2.2 quadrature_policy

discretization::quadrature::Policy serif::operators::backend::mfem::VolumeOptions::quadrature_policy {{}}

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.573 serif::mapping::VolumeQuadratureContext Struct Reference

#include <context.hpp>

Public Attributes

- mfem::DenseMatrix J_element_to_physical_inverse
- double mapping_determinate
- double physical_integration_weight

10.573.1 Member Data Documentation

10.573.1.1 J_element_to_physical_inverse

mfem::DenseMatrix serif::mapping::VolumeQuadratureContext::J_element_to_physical_inverse

10.573.1.2 mapping_determinate

double serif::mapping::VolumeQuadratureContext::mapping_determinate

10.573.1.3 physical_integration_weight

double serif::mapping::VolumeQuadratureContext::physical_integration_weight

The documentation for this struct was generated from the following file:

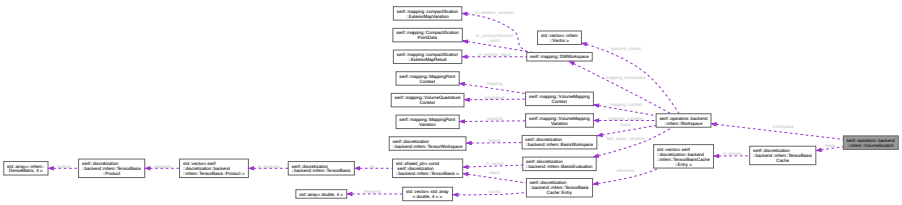
- src/include/serif/mapping/context.hpp

10.574 serif::operators::backend::mfem::VolumeScratch Struct Reference

Scratch belongs to graph callables and is never a dependency source.

#include <volume_data.hpp>

Collaboration diagram for serif::operators::backend::mfem::VolumeScratch:



Public Attributes

- Workspace workspace
- TensorBasisCache basis

10.574.1 Detailed Description

Scratch belongs to graph callables and is never a dependency source.

10.574.2 Member Data Documentation

10.574.2.1 basis

TensorBasisCache serif::operators::backend::mfem::VolumeScratch::basis

10.574.2.2 workspace

Workspace serif::operators::backend::mfem::VolumeScratch::workspace

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/volume_data.hpp

10.575 serif::dimensions::WithRespectTo< Quantity > Struct Template Reference

A type that is empty at runtime but allows for the developer to specify to the compiler what quantity the partial is with respect to.

#include <partial.hpp>

10.575.1 Detailed Description

template<ThermodynamicQuantityType Quantity>
struct serif::dimensions::WithRespectTo< Quantity >

A type that is empty at runtime but allows for the developer to specify to the compiler what quantity the partial is with respect to.

Template Parameters

Quantity	A quantity satisfying ThermodynamicQuantityType

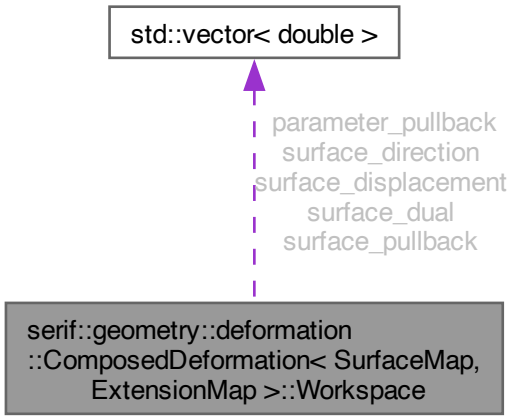
The documentation for this struct was generated from the following file:

- src/include/serif/dimensions/partial.hpp

10.576 serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace Struct Reference

#include <composition.hpp>

Collaboration diagram for serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace:



Public Attributes

- DeformationWorkspace< SurfaceMap > surface_workspace
- DeformationWorkspace< ExtensionMap > extension_workspace
- std::vector< double > surface_displacement
- std::vector< double > surface_direction
- std::vector< double > surface_dual
- std::vector< double > surface_pullback
- std::vector< double > parameter_pullback

10.576.1 Member Data Documentation

10.576.1.1 extension_workspace

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
DeformationWorkspace<ExtensionMap> serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace::extension_workspace

10.576.1.2 parameter_pullback

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
std::vector<double> serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace::parameter_pullback

10.576.1.3 surface_direction

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
std::vector<double> serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace::surface_direction

10.576.1.4 surface_displacement

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
std::vector<double> serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace::surface_displacement

10.576.1.5 surface_dual

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
std::vector<double> serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace::surface_dual

10.576.1.6 surface_pullback

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
std::vector<double> serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace::surface_pullback

10.576.1.7 surface_workspace

template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
DeformationWorkspace<SurfaceMap> serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace::surface_workspace

The documentation for this struct was generated from the following file:

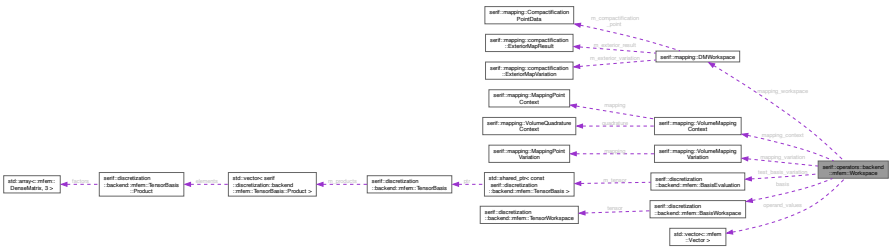
- src/include/serif/geometry/deformation/composition.hpp

10.577 serif::operators::backend::mfem::Workspace Struct Reference

Reusable scratch for applying the accepted element linearization.

#include <workspace.hpp>

Collaboration diagram for serif::operators::backend::mfem::Workspace:



Public Attributes

- mapping::DMWorkspace mapping_workspace {3}
- mapping::VolumeMappingContext mapping_context
- mapping::VolumeMappingVariation mapping_variation
- ::mfem::Vector element_values
- ::mfem::Vector element_result
- ::mfem::Vector quadrature_values
- ::mfem::Vector displacement_direction
- ::mfem::Vector full_result
- discretization::backend::mfem::BasisEvaluation test_basis_variation
- discretization::backend::mfem::BasisWorkspace basis

- ::mfem::Vector displacement_samples
- ::mfem::Vector displacement_value = ::mfem::Vector(3)
- ::mfem::DenseMatrix displacement_gradient {3}
- ::mfem::Vector test_values
- ::mfem::Vector reference_test_values
- ::mfem::Vector operand_dual
- ::mfem::Vector displacement_dual
- ::mfem::Vector test_variation_result
- std::vector<::mfem::Vector > operand_values

10.577.1 Detailed Description

Reusable scratch for applying the accepted element linearization.

10.577.2 Member Data Documentation

10.577.2.1 basis

discretization::backend::mfem::BasisWorkspace serif::operators::backend::mfem::Workspace::basis

10.577.2.2 displacement_direction

::mfem::Vector serif::operators::backend::mfem::Workspace::displacement_direction

10.577.2.3 displacement_dual

::mfem::Vector serif::operators::backend::mfem::Workspace::displacement_dual

10.577.2.4 displacement_gradient

::mfem::DenseMatrix serif::operators::backend::mfem::Workspace::displacement_gradient {3}

10.577.2.5 displacement_samples

::mfem::Vector serif::operators::backend::mfem::Workspace::displacement_samples

10.577.2.6 displacement_value

::mfem::Vector serif::operators::backend::mfem::Workspace::displacement_value = ::mfem::Vector(3)

10.577.2.7 element_result

::mfem::Vector serif::operators::backend::mfem::Workspace::element_result

10.577.2.8 element_values

::mfem::Vector serif::operators::backend::mfem::Workspace::element_values

10.577.2.9 full_result

::mfem::Vector serif::operators::backend::mfem::Workspace::full_result

10.577.2.10 mapping_context

mapping::VolumeMappingContext serif::operators::backend::mfem::Workspace::mapping_context

10.577.2.11 mapping_variation

mapping::VolumeMappingVariation serif::operators::backend::mfem::Workspace::mapping_variation

10.577.2.12 mapping_workspace

mapping::DMWorkspace serif::operators::backend::mfem::Workspace::mapping_workspace {3}

10.577.2.13 operand_dual

::mfem::Vector serif::operators::backend::mfem::Workspace::operand_dual

10.577.2.14 operand_values

std::vector<::mfem::Vector> serif::operators::backend::mfem::Workspace::operand_values

10.577.2.15 quadrature_values

::mfem::Vector serif::operators::backend::mfem::Workspace::quadrature_values

10.577.2.16 reference_test_values

::mfem::Vector serif::operators::backend::mfem::Workspace::reference_test_values

10.577.2.17 test_basis_variation

discretization::backend::mfem::BasisEvaluation serif::operators::backend::mfem::Workspace::test_basis_variation

10.577.2.18 test_values

::mfem::Vector serif::operators::backend::mfem::Workspace::test_values

10.577.2.19 test_variation_result

::mfem::Vector serif::operators::backend::mfem::Workspace::test_variation_result

The documentation for this struct was generated from the following file:

- src/include/serif/operators/backend/mfem/workspace.hpp

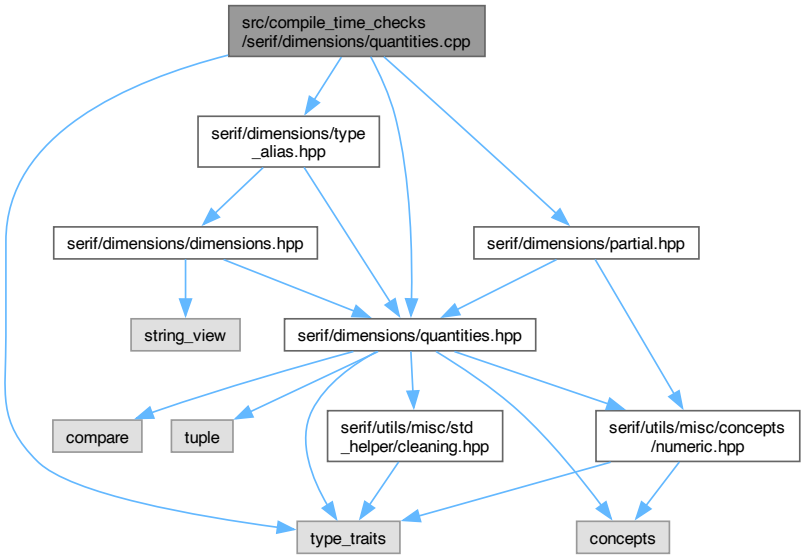
Chapter 11

File Documentation

11.1 docs/static/mainpage.md File Reference

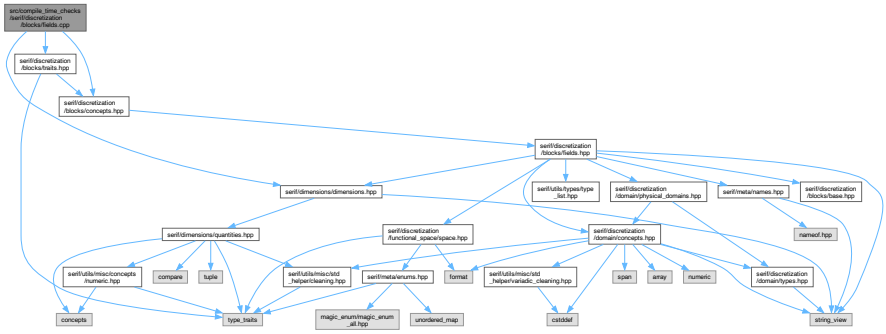
11.2 src/compile_time_checks/serif/dimensions/quantities.cpp File Reference

#include ”serif/dimensions/quantities.hpp”
#include ”serif/dimensions/type_alias.hpp”
#include ”serif/dimensions/partial.hpp”
#include <type_traits>
Include dependency graph for quantities.cpp:



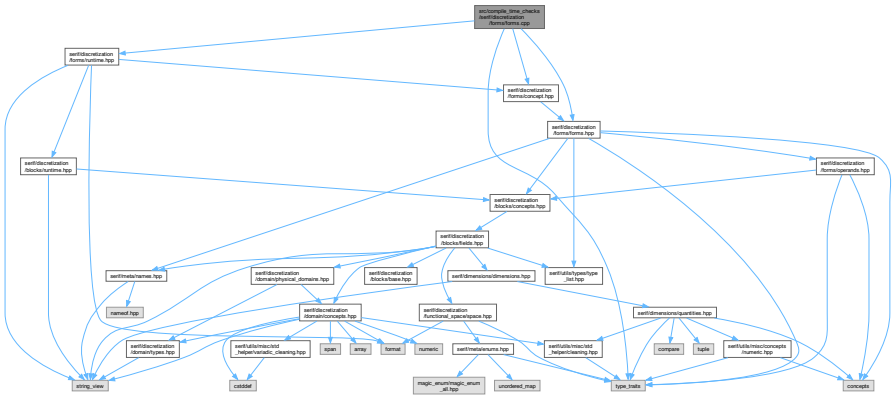
11.3 src/compile_time_checks/serif/discretization/blocks/fields.cpp File Reference

#include ”serif/discretization/blocks/concepts.hpp”
#include ”serif/discretization/blocks/traits.hpp”
#include ”serif/dimensions/dimensions.hpp”
Include dependency graph for fields.cpp:



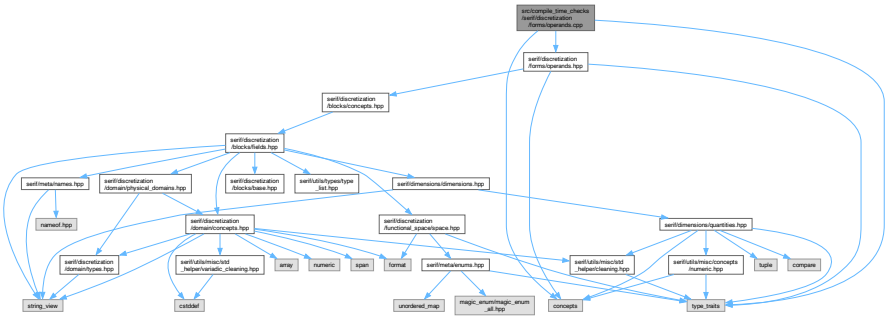
11.4 src/compile_time_checks/serif/discretization/forms/forms.cpp File Reference

#include <type_traits>
#include ”serif/discretization/forms/runtime.hpp”
#include ”serif/discretization/forms/concept.hpp”
#include ”serif/discretization/forms/forms.hpp”
Include dependency graph for forms.cpp:



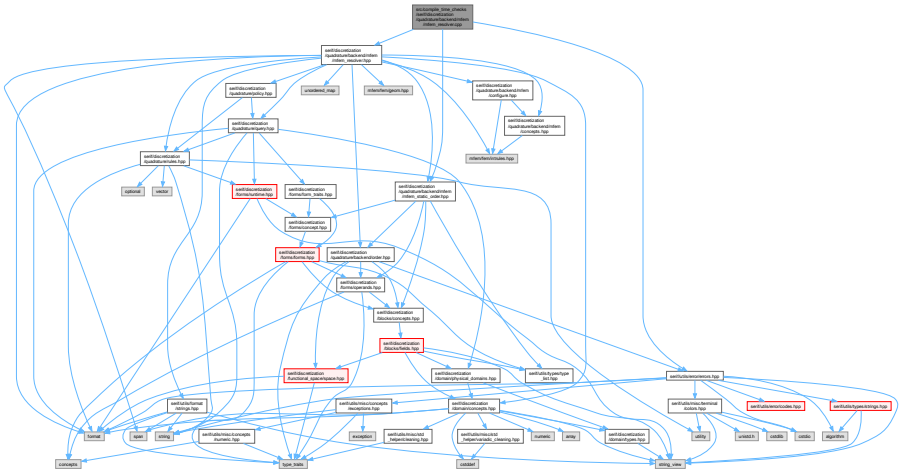
11.5 src/compile_time_checks/serif/discretization/forms/operands.cpp File Reference

#include "serif/discretization/forms/operands.hpp"
#include <type_traits>
#include <concepts>
Include dependency graph for operands.cpp:



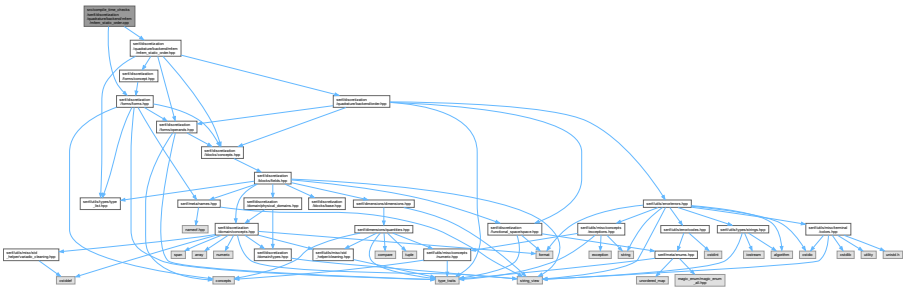
11.6 src/compile_time_↩ checks/serif/discretization/quadrature/backend/mfem/mfem_↩ resolver.cpp File Reference

#include "serif/discretization/quadrature/backend/mfem/mfem_resolver.hpp"
#include "serif/discretization/quadrature/backend/mfem/mfem_static_order.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for mfem_resolver.cpp:



11.7 src/compile_time_checks/serif/discretization/quadrature/backend/mfem/mfem_static_order.cpp File Reference

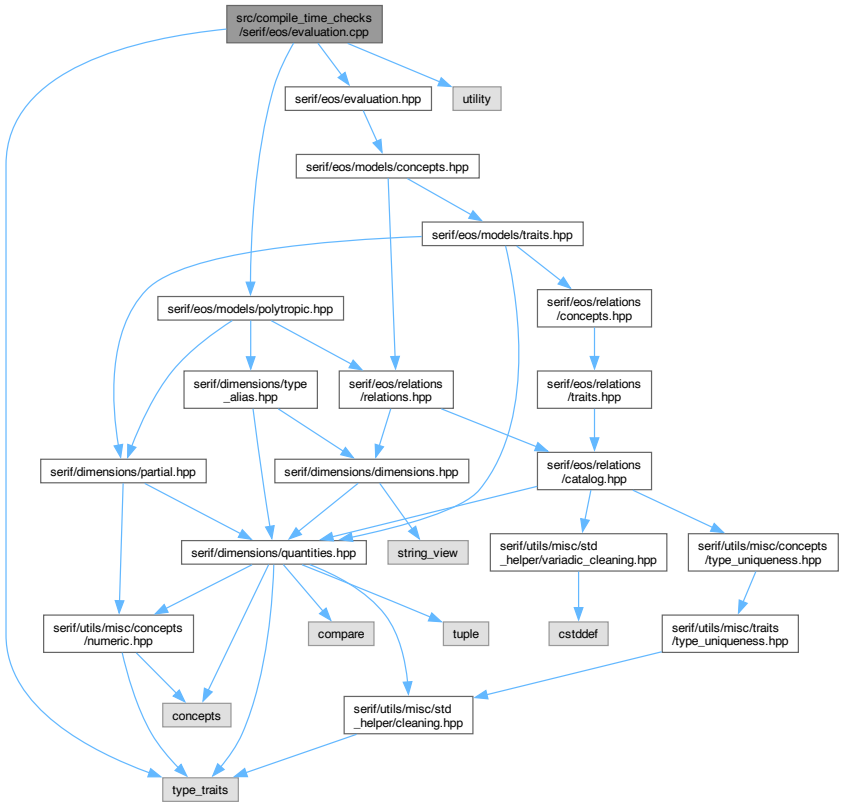
#include "serif/discretization/quadrature/backend/mfem/mfem_static_order.hpp"
#include "serif/discretization/forms/forms.hpp"
Include dependency graph for mfem_static_order.cpp:



11.8 src/compile_time_checks/serif/eos/evaluation.cpp File Reference

#include "serif/eos/evaluation.hpp"
#include "serif/eos/models/polytropic.hpp"
#include <type_traits>
#include <utility>

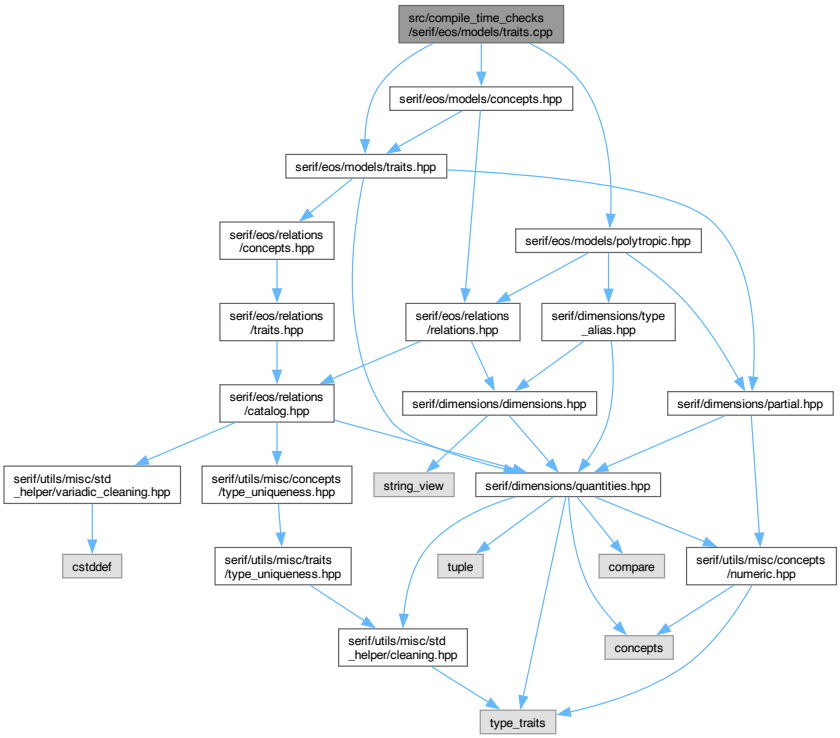
Include dependency graph for evaluation.cpp:



11.9 src/compile_time_checks/serif/eos/models/traits.cpp File Reference

#include ”serif/eos/models/traits.hpp”
#include ”serif/eos/models/concepts.hpp”
#include ”serif/eos/models/polytropic.hpp”

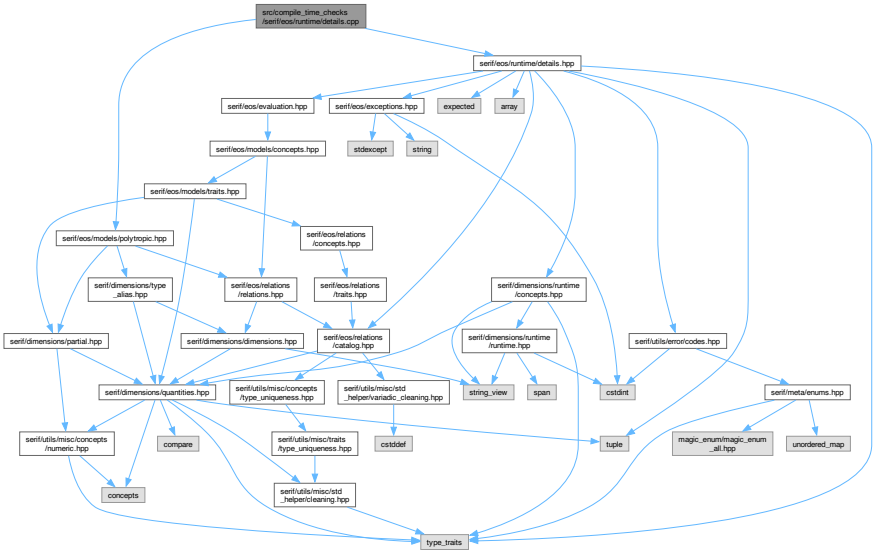
Include dependency graph for traits.cpp:



11.10 src/compile_time_checks/serif/eos/runtime/details.cpp File Reference

#include ”serif/eos/runtime/details.hpp”
#include ”serif/eos/models/polytropic.hpp”

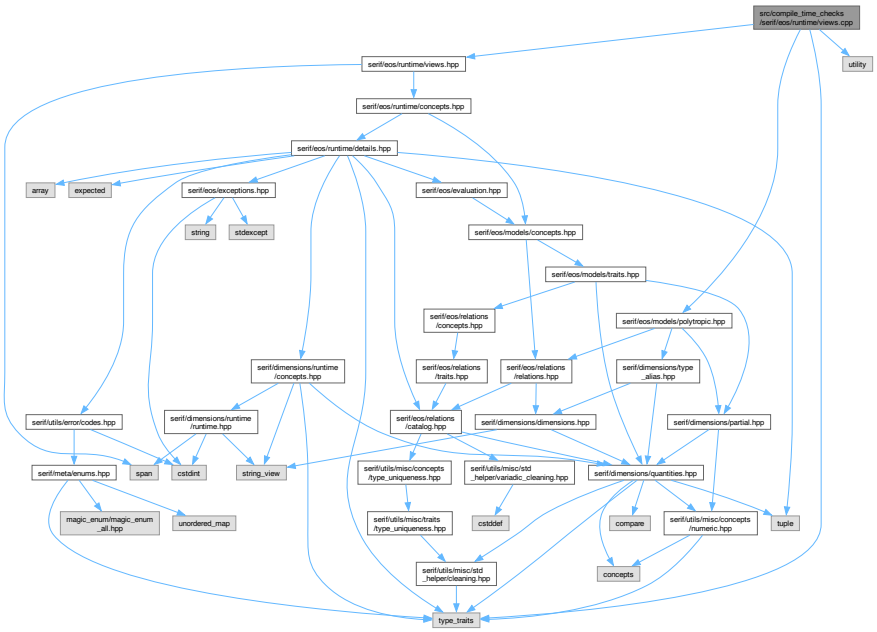
Include dependency graph for details.cpp:



11.11 src/compile_time_checks/serif/eos/runtime/views.cpp File Reference

#include ”serif/eos/runtime/views.hpp”
#include ”serif/eos/models/polytropic.hpp”
#include <type_traits>
#include <utility>

Include dependency graph for views.cpp:



Enumerations

- enum class SERiFErrorCode

11.11.1 Enumeration Type Documentation

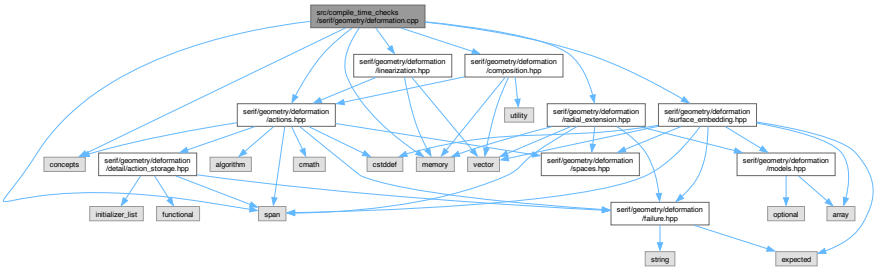
11.11.1.1 SERiFErrorCode

enum class serif::utils::errors::SERiFErrorCode : uint16_t [strong]

11.12 src/compile_time_checks/serif/geometry/deformation.cpp File Reference

#include <concepts>
#include <memory>
#include
#include "serif/geometry/deformation/actions.hpp"
#include "serif/geometry/deformation/composition.hpp"
#include "serif/geometry/deformation/linearization.hpp"
#include "serif/geometry/deformation/radial_extension.hpp"

#include ”serif/geometry/deformation/surface_embedding.hpp”
Include dependency graph for deformation.cpp:



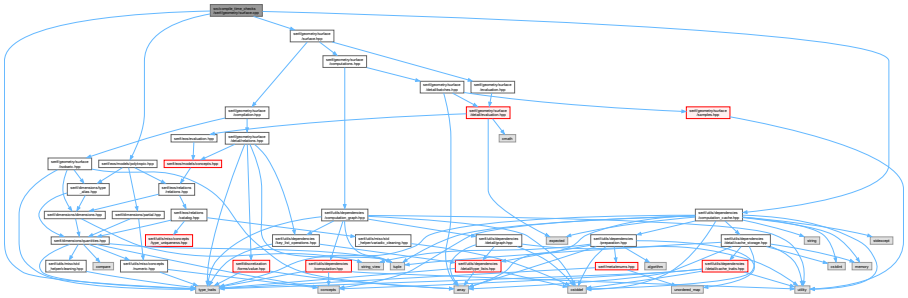
Concepts

- concept @230372075125117112146216022050257331177137107327::CanCompose

  Equal vector lengths cannot substitute for matching physical meanings at a composition boundary.

11.13 src/compile_time_checks/serif/geometry/surface.cpp File Reference

#include <type_traits>
#include ”serif/eos/models/polytropic.hpp”
#include ”serif/geometry/surface/surface.hpp”
#include ”serif/utils/dependencies/computation_cache.hpp”
Include dependency graph for surface.cpp:

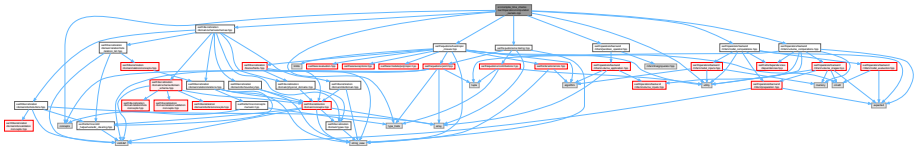


Concepts

- concept @247006277077211216153350341142357337161001026143::SelectsCarrier
- concept @247006277077211216153350341142357337161001026143::HasSurfaceRelation

11.14 src/compile_time_checks/serif/operators/computation_owners.cpp File Reference

#include <concepts>
#include <type_traits>
#include <utility>
#include "serif/discretization/domain/schema/schemas.hpp"
#include "serif/equations/barotropic_closure.hpp"
#include "serif/equations/centering.hpp"
#include "serif/operators/backend/mfem/jacobian_operator.hpp"
#include "serif/operators/backend/mfem/volume_computations.hpp"
#include "serif/operators/backend/mfem/nodal_computations.hpp"
Include dependency graph for computation_owners.cpp:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Concepts

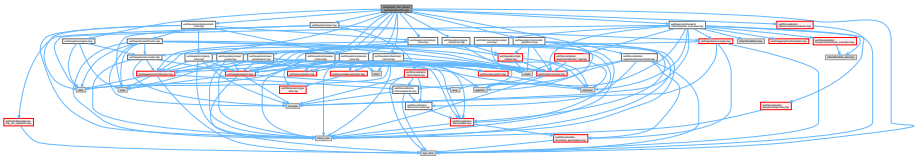
- concept serif::operators::backend::mfem::ClosureResult
- concept serif::operators::backend::mfem::ClosureInputsAccept

Typedefs

- using serif::operators::backend::mfem::Schema = discretization::domain::schema::CoreEnvelopeVacuumDomainSchema
- using serif::operators::backend::mfem::Closure = equations::BarotropicClosure<eos::models::Polytrope>
- using serif::operators::backend::mfem::Centering = equations::CenteringConstraint
- using serif::operators::backend::mfem::Density = discretization::forms::Operand<discretization::blocks::density::mass>
- using serif::operators::backend::mfem::DensityBinding = discretization::backend::mfem::TermBinding<Closure::Test, Schema>
- using serif::operators::backend::mfem::EnthalpyBinding = discretization::backend::mfem::TermBinding<discretization::b Schema>
- using serif::operators::backend::mfem::DisplacementBinding = discretization::backend::mfem::TermBinding<Centering:: Schema>
- using serif::operators::backend::mfem::Vector = ::mfem::Vector
- using serif::operators::backend::mfem::VolumeCoefficients
- using serif::operators::backend::mfem::VolumeMesh = VolumeDiscretization<Closure::Test, Schema>
- using serif::operators::backend::mfem::VolumeInputs
- using serif::operators::backend::mfem::VolumeCache
- using serif::operators::backend::mfem::NodalCoefficients
- using serif::operators::backend::mfem::NodalMesh = decltype(make_point_discretization<Centering>(std↩ ::declval<const DisplacementBinding&>()))
- using serif::operators::backend::mfem::NodalCache

11.15 src/compile_time_checks/serif/operators/plan.cpp File Reference

#include <type_traits>
#include "serif/equations/displacement_force.hpp"
#include "serif/equations/barotropic_closure.hpp"
#include "serif/equations/hydrostatic_equilibrium.hpp"
#include "serif/operators/plan.hpp"
#include "serif/equations/registry.hpp"
#include "serif/equations/mixed_gravity.hpp"
#include "serif/equations/gravity_force.hpp"
#include "serif/equations/rotational_force.hpp"
#include "serif/equations/pressure_force.hpp"
#include "serif/equations/pressure_surface.hpp"
#include "serif/operators/backend/mfem/operand_evaluation.hpp"
#include "serif/discretization/quadrature/backend/order.hpp"
#include "serif/discretization/domain/schema/schemas.hpp"
#include "serif/equations/mass_normalization.hpp"
#include "serif/equations/angular_momentum.hpp"
Include dependency graph for plan.cpp:



Classes

- struct serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >
- struct serif::operators::GradientRequest
- struct serif::operators::BadContributionRow
- struct serif::operators::BadContributionCollection
- struct serif::operators::NestedBadCollection
- class serif::operators::BarotropicHydrostaticEquilibrium
- class serif::operators::EquationRegistry< EquationTs >
- struct serif::operators::OperandValues< OperandsT, OutputT >

Namespaces

- namespace serif
- namespace serif::operators

Concepts

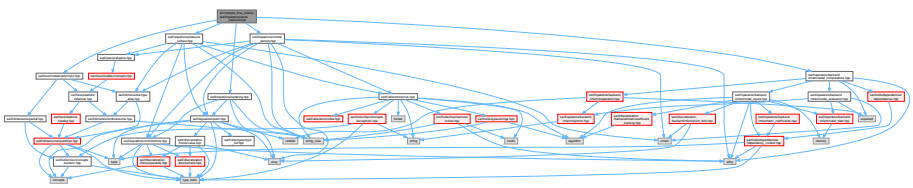
- concept serif::operators::CanSum
- concept serif::operators::CanSubtract

Typedefs

- using serif::operators::Closure = equations::BarotropicClosure<eos::models::Polytrope>
- using serif::operators::Density = discretization::blocks::density::mass
- using serif::operators::Enthalpy = discretization::blocks::enthalpy::specific
- using serif::operators::Schema = discretization::domain::schema::CoreEnvelopeVacuumDomainSchema
- using serif::operators::SurfacePressure = equations::PressureSurfaceConstraint<eos::models::Polytrope>
- using serif::operators::DensityValue = discretization::forms::Operand<Density>
- using serif::operators::EnthalpyValue = discretization::forms::Operand<Enthalpy>
- using serif::operators::EnthalpyGradient = discretization::forms::Operand<Enthalpy, discretization::forms::Operation::G
- using serif::operators::GravityValue = discretization::forms::Operand<discretization::blocks::gravity::gradient>
- using serif::operators::GravityDivergence = discretization::forms::Operand<discretization::blocks::gravity::gradient, discretization::forms::Operation::Divergence>
- using serif::operators::Pressure = equations::PressureForce<eos::models::Polytrope>
- using serif::operators::DisplacementDivergence = discretization::forms::Operand<discretization::blocks::displacement::ve discretization::forms::Operation::Divergence>
- using serif::operators::Displacement = discretization::blocks::displacement::vector
- using serif::operators::DisplacementValue = discretization::forms::Operand<Displacement>
- using serif::operators::Potential = discretization::blocks::gravity::potential
- using serif::operators::MixedSamples = utils::types::TypeList<GravityValue, GravityDivergence, DensityValue>
- using serif::operators::Mechanical = equations::DisplacementForce<eos::models::Polytrope>
- using serif::operators::MechanicalDifference = equations::EquationDifference<Pressure, equations::EquationSum<equat equations::RotationalForce>>
- using serif::operators::ScaledMechanical = equations::EquationScale<MechanicalDifference>
- using serif::operators::ScaledPoisson = equations::EquationScale<equations::PoissonBalance>
- using serif::operators::Mass = equations::MassNormalization
- using serif::operators::Constant = discretization::blocks::barotropic_constant::mass_normalization
- using serif::operators::Omega = discretization::blocks::rotation::angular_velocity
- using serif::operators::Momentum = equations::AngularMomentum

11.16 src/compile_time_checks/serif/operators/serial_backend.cpp File Reference

#include "serif/eos/models/polytropic.hpp"
#include "serif/equations/central_density.hpp"
#include "serif/equations/centering.hpp"
#include "serif/equations/pressure_surface.hpp"
#include "serif/operators/backend/mfem/nodal_computations.hpp"
Include dependency graph for serial_backend.cpp:



Namespaces

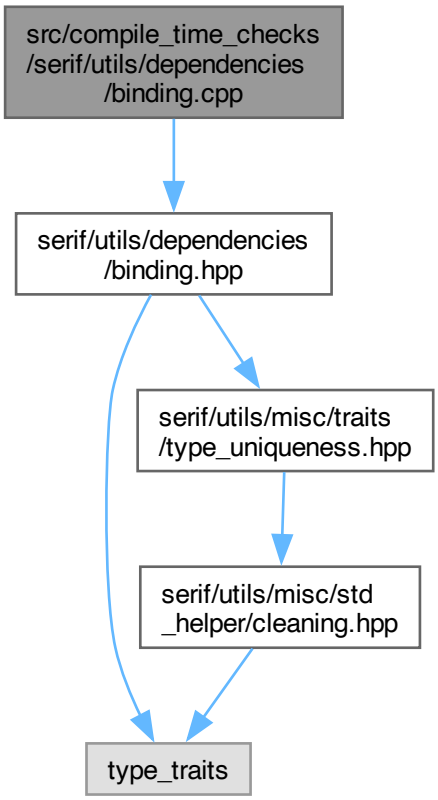
- namespace serif
- namespace serif::operators

Typedefs

- using serif::operators::CentralDensity = equations::CentralDensityConstraint<eos::models::Polytrope>

11.17 src/compile_time_checks/serif/utils/dependencies/binding.cpp File Reference

#include "serif/utils/dependencies/binding.hpp"
Include dependency graph for binding.cpp:

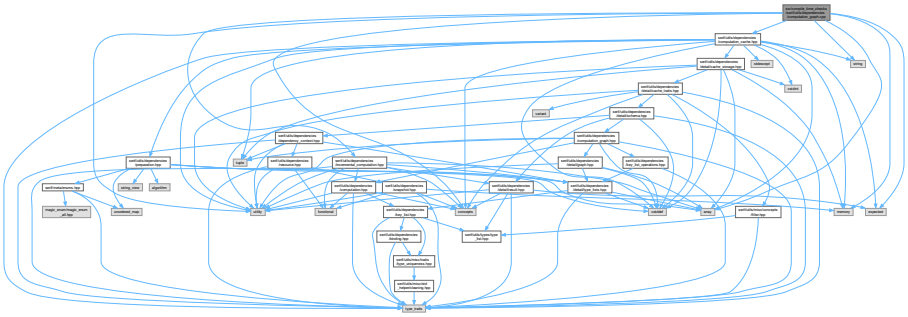


Concepts

- concept @1200271712111733323463461531470020022660652232 66::HasBoundType
- concept @1200271712111733323463461531470020022660652232 66::HasBoundTypeMember

11.18 src/compile_time_checks/serif/utils/dependencies/computation_↩ graph.cpp File Reference

#include ”serif/utils/dependencies/computation_cache.hpp”
#include ”serif/utils/dependencies/incremental_computation.hpp”
#include <array>
#include <concepts>
#include <expected>
#include <memory>
#include <string>
#include <utility>
Include dependency graph for computation_graph.cpp:



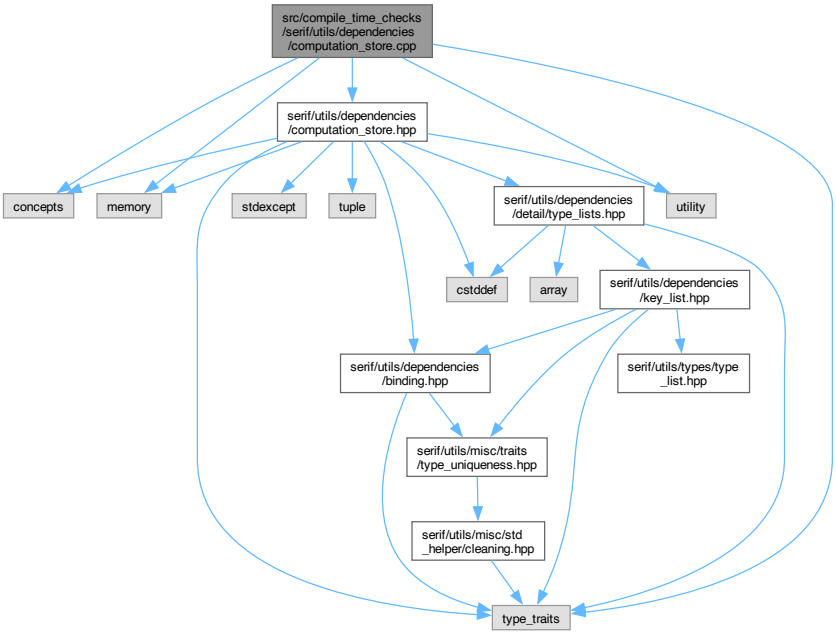
Concepts

- concept @2000743173031670560630642560700462030540640522 14::CanRequest
- concept @2000743173031670560630642560700462030540640522 14::CanInspectChange
- concept @2000743173031670560630642560700462030540640522 14::CanMakeComputation
- concept @2000743173031670560630642560700462030540640522 14::CanMakeCache
- concept @2000743173031670560630642560700462030540640522 14::CanMakeFallibleCache

11.19 src/compile_time_checks/serif/utils/dependencies/computation_↩ store.cpp File Reference

#include ”serif/utils/dependencies/computation_store.hpp”
#include <concepts>
#include <memory>
#include <type_traits>

#include <utility>
Include dependency graph for computation_store.cpp:



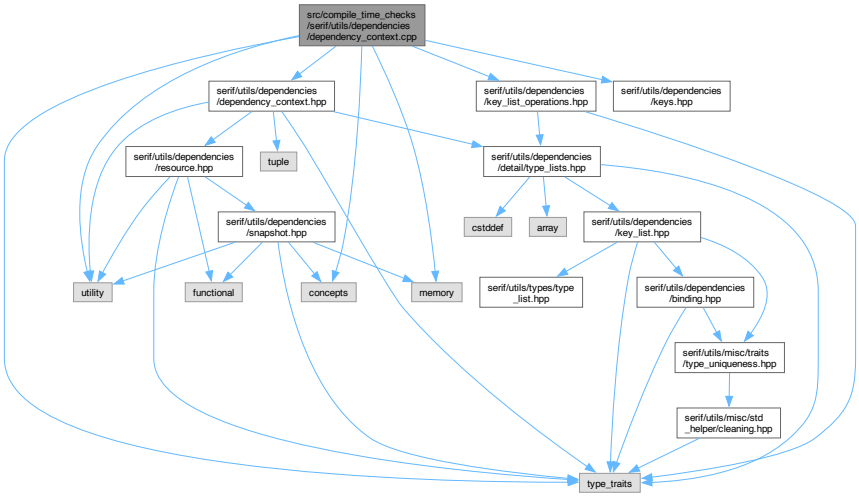
Concepts

- concept @347145241034102141072004357171274360230346064356::CanOwn
- concept @347145241034102141072004357171274360230346064356::CanMakeStore
- concept @347145241034102141072004357171274360230346064356::CanRead
- concept @347145241034102141072004357171274360230346064356::CanReadEntry

11.20 src/compile_time_checks/serif/utils/dependencies/dependency_context.cpp File Reference

#include "serif/utils/dependencies/dependency_context.hpp"
#include "serif/utils/dependencies/key_list_operations.hpp"
#include "serif/utils/dependencies/keys.hpp"
#include <concepts>
#include <memory>
#include <type_traits>

#include <utility>
Include dependency graph for dependency_context.cpp:



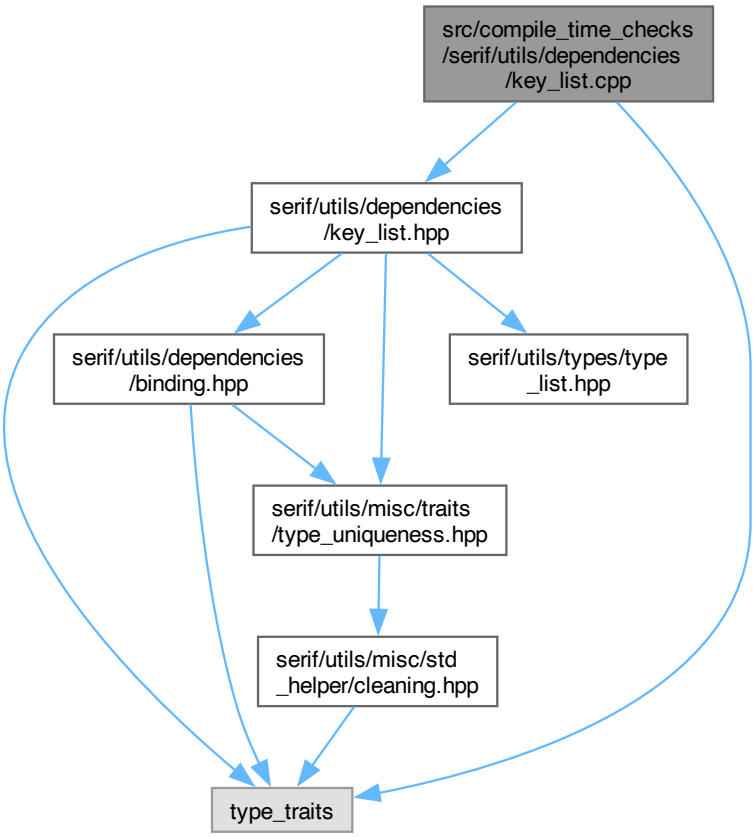
Concepts

- concept @323172327364102235210333337057065015152354331343::CanMakeContext
- concept @323172327364102235210333337057065015152354331343::CanRead
- concept @323172327364102235210333337057065015152354331343::CanSelect
- concept @323172327364102235210333337057065015152354331343::CanBorrow

11.21 src/compile_time_checks/serif/utils/dependencies/key_list.cpp File Reference

#include "serif/utils/dependencies/key_list.hpp"
#include <type_traits>

Include dependency graph for key_list.cpp:



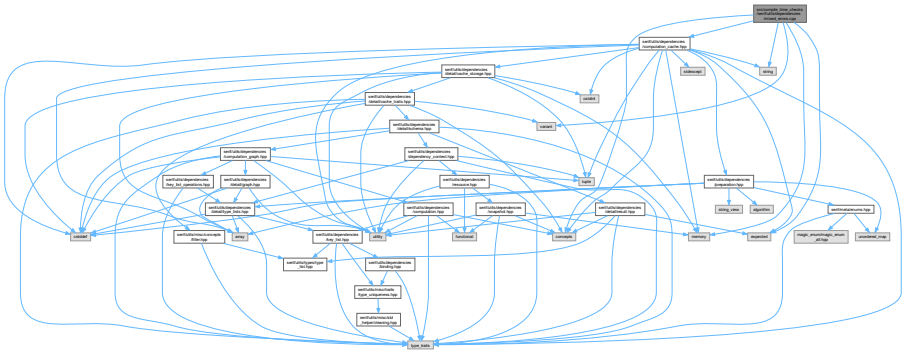
Concepts

- concept @252166143035264314021214246026055041120163322041::CanResolve

11.22 src/compile_time_checks/serif/utils/dependencies/mixed_errors.cpp File Reference

#include <concepts>
#include <expected>
#include <memory>
#include <string>
#include <variant>

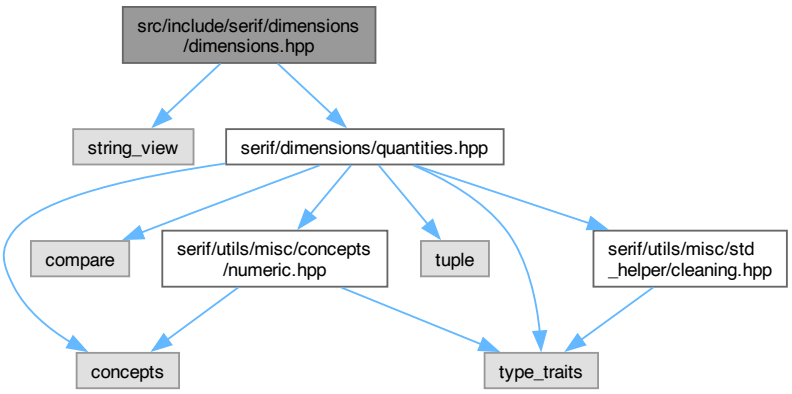
#include ”serif/utils/dependencies/computation_cache.hpp”
Include dependency graph for mixed_errors.cpp:



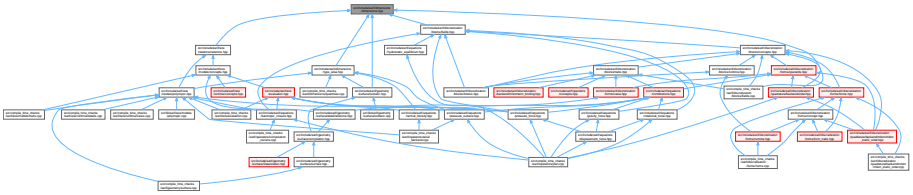
11.23 src/include/serif/dimensions/dimensions.hpp File Reference

File containing prue dimensions. Note that this is not intended to be (nor does it have the capability to be) a general dimensional analysis library. There is no understanding here that, for example, a dimension of length^2 is an area. Rather, the purpose of this is to force developers to be explicit about what values they are passing to what functions. Further, most of this system is built to be compile time constant. This means that the compiler can still apply many optimizations. Note that the dimensions here do not contain a value, rather these are just the concept of a dimension. To store a value you must use a Quantity and QuantityValue.

#include <string_view>
#include ”serif/dimensions/quantities.hpp”
Include dependency graph for dimensions.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::dimensions::Dimensionless
  Non dimensional type.
- struct serif::dimensions::Mass
  Dimension of mass.
- struct serif::dimensions::Length
  Dimension of Length.
- struct serif::dimensions::Time
  Dimension of Time.
- struct serif::dimensions::Area
  Dimension of Area.
- struct serif::dimensions::Volume
  Dimension of Volume.
- struct serif::dimensions::MassDensity
  Dimension of Mass Density.
- struct serif::dimensions::NumberDensity
  Dimension of Number Density.
- struct serif::dimensions::Pressure
  Dimension of Pressure.
- struct serif::dimensions::Temperature
  Dimension of Temperature.
- struct serif::dimensions::Entropy
  Dimension of Entropy.
- struct serif::dimensions::SpecificEntropy
  Dimension of Specific Entropy.
- struct serif::dimensions::ChemicalPotential
  Dimension of Chemical Potential.
- struct serif::dimensions::Energy
  Dimension of Energy.
- struct serif::dimensions::InternalEnergy
  Dimension of Internal Energy.
- struct serif::dimensions::SpecificEnergy
  Dimension of Specific Energy.
- struct serif::dimensions::SpecificInternalEnergy
  Dimension of Specific Internal Energy.
- struct serif::dimensions::SpecificEnthalpy

Dimension of Specific Enthalpy.

- struct serif::dimensions::EnergyDensity

  Dimension of Energy Density.
- struct serif::dimensions::GravitationalPotential

  Dimension of Gravitational Potential.
- struct serif::dimensions::Velocity

  Dimension of Velocity.
- struct serif::dimensions::Acceleration

  Dimension of Acceleration.
- struct serif::dimensions::Frequency

  Dimension of Frequency.
- struct serif::dimensions::AngularVelocity

  Dimension of Angular Velocity.
- struct serif::dimensions::Momentum

  Dimension of Momentum.
- struct serif::dimensions::AngularMomentum

  Dimension of Angular Momentum.
- struct serif::dimensions::MomentOfInertia

  Dimension of Moment Of Inertia.
- struct serif::dimensions::Force

  Dimension of Force.
- struct serif::dimensions::Torque

  Dimension of Torque.
- struct serif::dimensions::Power

  Dimension of Power.
- struct serif::dimensions::Luminosity

  Dimension of Luminosity.
- struct serif::dimensions::MassFlowRate

  Dimension of Mass Flow Rate.
- struct serif::dimensions::Opacity

  Dimension of Opacity.
- struct serif::dimensions::DynamicViscosity

  Dimension of Dynamic Viscosity.
- struct serif::dimensions::KinematicViscosity

  Dimension of Kinematic Viscosity.
- struct serif::dimensions::MagneticFluxDensity

  Dimension of Magnetic Flux Density.

Namespaces

- namespace serif
- namespace serif::dimensions

11.23.1 Detailed Description

File containing prue dimensions. Note that this is not intended to be (nor does it have the capability to be) a general dimensional analysis library. There is no understanding here that, for example, a dimension of length^2 is an area. Rather, the purpose of this is to force developers to be explicit about what values they are passing to what functions. Further, most of this system is built to be compile time constant. This means that the compiler can still apply many optimizations. Note that the dimensions here do not contain a value, rather these are just the concept of a dimension. To store a value you must use a Quantity and QuantityValue.

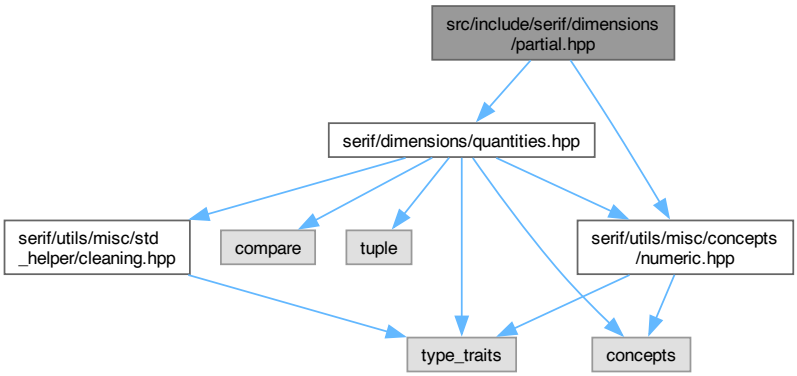
Author

Emily M. Boudreaux

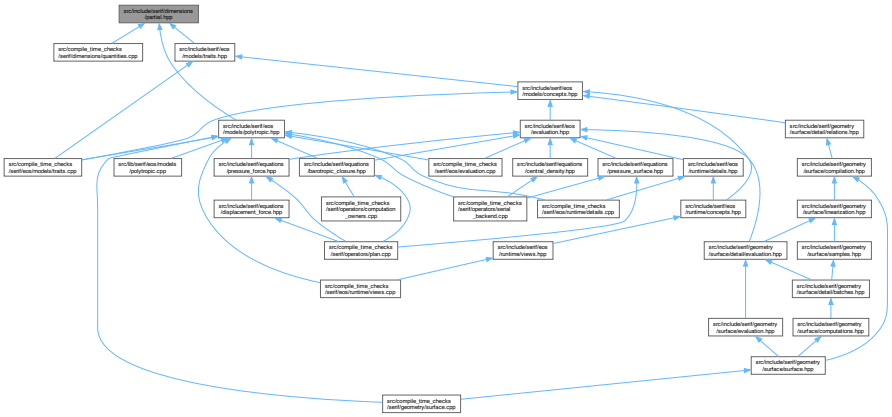
11.24 src/include/serif/dimensions/partial.hpp File Reference

Compile time tools to represent a partial derivative of one quantity wrt. another quantity. Note that this system is not responsible for computing partials or in fact derivatives of any kind. Rather, it is responsible for forcing the developer — or compiler at least — to ask for a specific, explicit, partial and have to spell out what that is. That in turn gives the compiler enough information to do a lot of downstream things. A motivating example would be automatic verification of the provided EOS against all partials that might be needed to ensure they are all exported by the EOS.

#include "serif/dimensions/quantities.hpp"
#include "serif/utils/misc/concepts/numeric.hpp"
Include dependency graph for partial.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::dimensions::WithRespectTo< Quantity >

  A type that is empty at runtime but allows for the developer to specify to the compiler what quantity the partial is with respect to.
- class serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >

  Class to represent a compile time derivative relation between two quantities. Note that this class does not, in fact cannot, compute any derivatives itself. Rather, this is a way for the compiler to understand higher level mathematical relations. Unless you are really in the weeds of the development team you will likely not need to interact with this directly. Further note that this class does include (unlike the more basic dimension system) some basic dimensional analysis. Specifically, there are some overloaded operators that allow for certain specializations of PartialDerivative to operate on one and other conditioned on what their input and output types are. This does not do any thing like convert area to length^2. However, it will do more basic things like ensure that you can only add two partials which agree on their dimensions exactly. Or that while you can multiple a partial by any scalar, you cannot multiply arbitrary partials.

Namespaces

- namespace serif
- namespace serif::dimensions

11.24.1 Detailed Description

Compile time tools to represent a partial derivative of one quantity wrt. another quantity. Note that this system is not responsible for computing partials or in fact derivatives of any kind. Rather, it is responsible for forcing the developer — or compiler at least — to ask for a specific, explicit, partial and have to spell out what that is. That in turn gives the compiler enough information to do a lot of downstream things. A motivating example would be automatic verification of the provided EOS against all partials that might be needed to ensure they are all exported by the EOS.

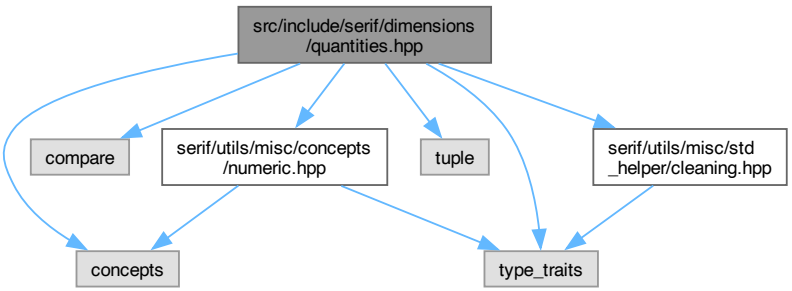
Author

Emily M. Boudreaux

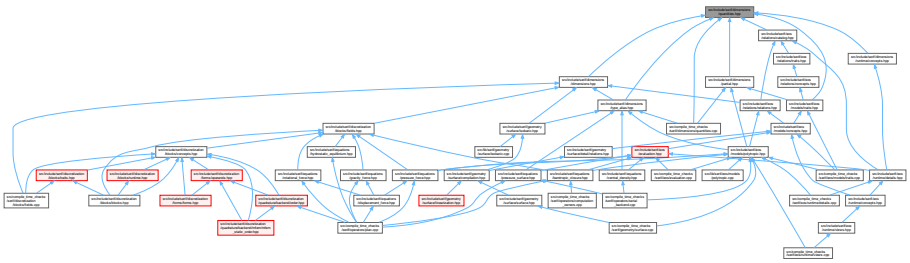
11.25 src/include/serif/dimensions/quantities.hpp File Reference

This file contains the code to associate numeric data to a dimension. We call this a Quantity and a Quantity↩ Value.

#include <type_traits>
#include <compare>
#include <concepts>
#include <tuple>
#include "serif/utils/misc/std_helper/cleaning.hpp"
#include "serif/utils/misc/concepts/numeric.hpp"
Include dependency graph for quantities.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::dimensions::PhysicalQuantity
- struct serif::dimensions::ThermodynamicQuantity
- class serif::dimensions::QuantityValue< Quantity >

  This is the non derivative equivalent to PartialDerivative. This associates a dimension with a value and, if that value can be determined at compile time, can be decomposed to a transparent double through .value()
- struct serif::dimensions::IsQuantityValue< Candidate >
- struct serif::dimensions::IsQuantityValue< QuantityValue< Quantity > >
- struct serif::dimensions::QuantityOf< QuantityValue< Quantity > >
- struct serif::dimensions::QuantityList< Quantities >
- struct serif::dimensions::QuantityAt< Index, QuantityList< Quantities... > >

Namespaces

- namespace serif
- namespace serif::dimensions

Concepts

- concept serif::dimensions::PhysicalQuantityType

  Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity. Note all ThermodynamicQuantities will also satisfy this concept.
- concept serif::dimensions::ThermodynamicQuantityType

  Is compile-time true of the QuantityCandidate can have any const / volatile stripped and if its derived from PhysicalQuantity (both enforced through checking PhysicalQuantityType) and then that is is also derived from ThermodynamicQuantity. Note not all physical quanties will satisfy this.
- concept serif::dimensions::QuantityValueType

  Turn those compile time structs into a concept that can be used to constrain a template parameter.
- concept serif::dimensions::ThermodynamicQuantityValueType

Typedefs

- template<QuantityValueType Value>
  using serif::dimensions::QuantityOfT = typename QuantityOf<NeitherConstNorVolatileReference<Value>>::Type
- template<class T>
  using serif::dimensions::NeitherConstNorVolatile
- template<class T>
  using serif::dimensions::NeitherConstNorVolatileReference

11.25.1 Detailed Description

This file contains the code to associate numeric data to a dimension. We call this a Quantity and a QuantityValue.

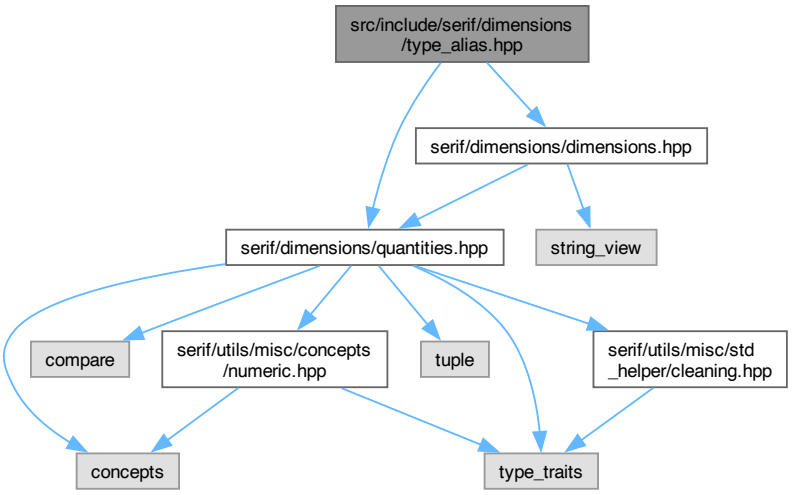
Author

Emily M. Boudreaux

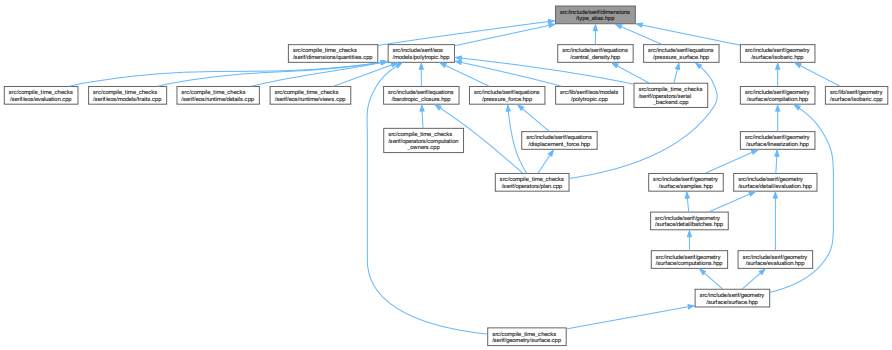
11.26 src/include/serif/dimensions/type_alias.hpp File Reference

This file contains type aliases for the dimensions and quantities used in the discretization. This is solely useful in so far as it means we do not need to type out the longer QuantityValue<Dimension> type for each of the dimensions and quantities used in the discretization. This is a convenience file that is used to make the code more readable and maintainable.

#include ”serif/dimensions/quantities.hpp”
#include ”serif/dimensions/dimensions.hpp”
Include dependency graph for type_alias.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::dimensions

Typedefs

- using serif::dimensions::DimensionlessValue = QuantityValue<Dimensionless>

  A dimensionless quantity, used for quantities that do not have a physical dimension (e.g., a pure number).

- using serif::dimensions::MassValue = QuantityValue<Mass>

  A quantity representing mass.
- using serif::dimensions::LengthValue = QuantityValue<Length>

  A quantity representing length.
- using serif::dimensions::TimeValue = QuantityValue<Time>

  A quantity representing time.
- using serif::dimensions::AreaValue = QuantityValue<Area>

  A quantity representing area.
- using serif::dimensions::VolumeValue = QuantityValue<Volume>

  A quantity representing volume.
- using serif::dimensions::DensityValue = QuantityValue<MassDensity>

  A quantity representing mass density.
- using serif::dimensions::NumberDensityValue = QuantityValue<NumberDensity>

  A quantity representing number density.
- using serif::dimensions::PressureValue = QuantityValue<Pressure>

  A quantity representing pressure.
- using serif::dimensions::TemperatureValue = QuantityValue<Temperature>

  A quantity representing temperature.
- using serif::dimensions::EntropyValue = QuantityValue<Entropy>

  A quantity representing entropy.
- using serif::dimensions::SpecificEntropyValue = QuantityValue<SpecificEntropy>

  A quantity representing specific entropy.
- using serif::dimensions::ChemicalPotentialValue = QuantityValue<ChemicalPotential>

  A quantity representing chemical potential.
- using serif::dimensions::EnergyValue = QuantityValue<Energy>

  A quantity representing energy..
- using serif::dimensions::InternalEnergyValue = QuantityValue<InternalEnergy>

  A quantity representing internal energy.
- using serif::dimensions::SpecificEnergyValue = QuantityValue<SpecificEnergy>

  A quantity representing specific energy.
- using serif::dimensions::SpecificInternalEnergyValue = QuantityValue<SpecificInternalEnergy>

  A quantity representing specific internal energy.
- using serif::dimensions::SpecificEnthalpyValue = QuantityValue<SpecificEnthalpy>

  A quantity representing specific enthalpy.
- using serif::dimensions::EnergyDensityValue = QuantityValue<EnergyDensity>

  A quantity representing energy density.
- using serif::dimensions::GravitationalPotentialValue = QuantityValue<GravitationalPotential>

  A quantity representing gravitational potential.
- using serif::dimensions::VelocityValue = QuantityValue<Velocity>

  A quantity representing velocity.
- using serif::dimensions::AccelerationValue = QuantityValue<Acceleration>

  A quantity representing acceleration.
- using serif::dimensions::FrequencyValue = QuantityValue<Frequency>

  A quantity representing frequency.
- using serif::dimensions::AngularVelocityValue = QuantityValue<AngularVelocity>

  A quantity representing angular velocity.
- using serif::dimensions::MomentumValue = QuantityValue<Momentum>

A quantity representing momentum.

- using serif::dimensions::AngularMomentumValue = QuantityValue<AngularMomentum>

  A quantity representing angular momentum.
- using serif::dimensions::MomentOfInertiaValue = QuantityValue<MomentOfInertia>

  A quantity representing moment of inertia.
- using serif::dimensions::ForceValue = QuantityValue<Force>

  A quantity representing force..
- using serif::dimensions::TorqueValue = QuantityValue<Torque>

  A quantity representing torque.
- using serif::dimensions::PowerValue = QuantityValue<Power>

  A quantity representing power.
- using serif::dimensions::LuminosityValue = QuantityValue<Luminosity>

  A quantity representing luminosity.
- using serif::dimensions::MassFlowRateValue = QuantityValue<MassFlowRate>

  A quantity representing mass flow rate.
- using serif::dimensions::OpacityValue = QuantityValue<Opacity>

  A quantity representing opacity.
- using serif::dimensions::DynamicViscosityValue = QuantityValue<DynamicViscosity>

  A quantity representing dynamic viscosity.
- using serif::dimensions::KinematicViscosityValue = QuantityValue<KinematicViscosity>

  A quantity representing kinematic viscosity.
- using serif::dimensions::MagneticFluxDensityValue = QuantityValue<MagneticFluxDensity>

  A quantity representing magnetic flux density.

11.26.1 Detailed Description

This file contains type aliases for the dimensions and quantities used in the discretization. This is solely useful in so far as it means we do not need to type out the longer QuantityValue<Dimension> type for each of the dimensions and quantities used in the discretization. This is a convenience file that is used to make the code more readable and maintainable.

Author

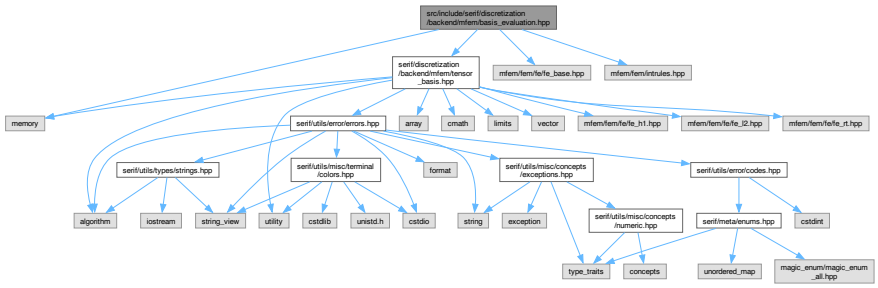
Emily M. Boudreaux

11.27 src/include/serif/discretization/backend/mfem/basis_evaluation.hpp File Reference

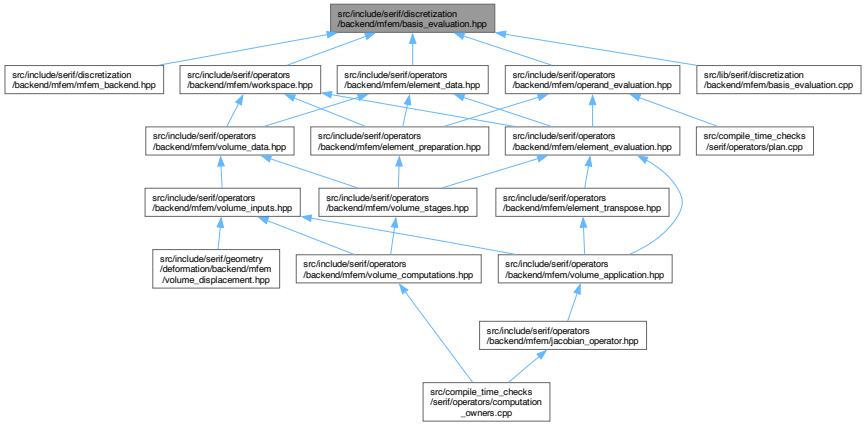
Basis evaluation, pointwise mapping, and transpose application using dense or tensor-product representations.

#include <memory>
#include "mfem/fem/fe/fe_base.hpp"
#include "mfem/fem/intrules.hpp"

#include ”serif/discretization/backend/mfem/tensor_basis.hpp”
Include dependency graph for basis_evaluation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::backend::mfem::BasisWorkspace

  Reusable scratch storage for basis contractions, reducing repeated allocations.
- class serif::discretization::backend::mfem::BasisEvaluation

  Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::backend
- namespace serif::discretization::backend::mfem

11.27.1 Detailed Description

Basis evaluation, pointwise mapping, and transpose application using dense or tensor-product representations.

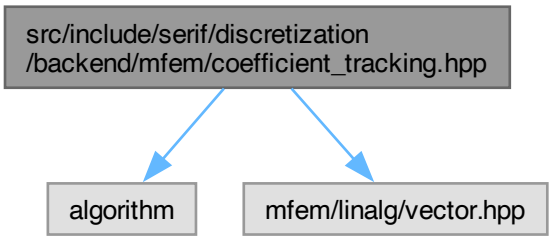
Author

Emily M. Boudreaux

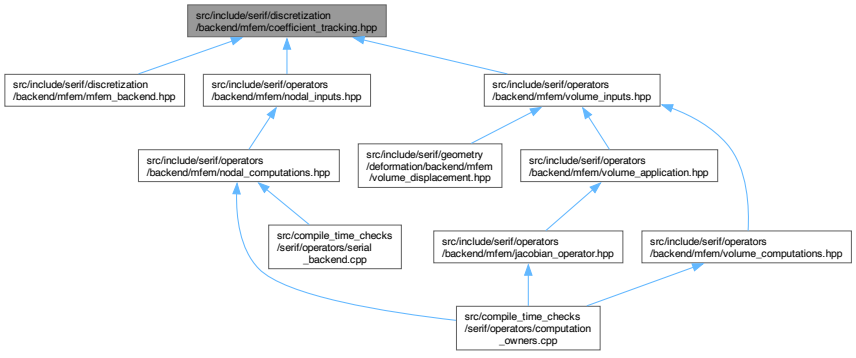
11.28 src/include/serif/discretization/backend/mfem/coefficient_↩ tracking.hpp File Reference

Snapshot and comparison policy for MFEM coefficient views.

#include <algorithm>
#include "mfem/linalg/vector.hpp"
Include dependency graph for coefficient_tracking.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::backend::mfem::CoefficientTracking

  Track coefficient values using an owned snapshot and the identity of their binding.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::backend
- namespace serif::discretization::backend::mfem

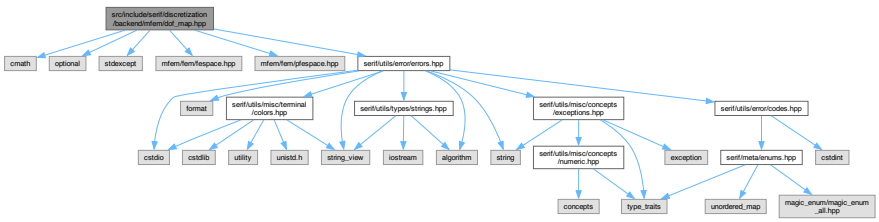
11.28.1 Detailed Description

Snapshot and comparison policy for MFEM coefficient views.

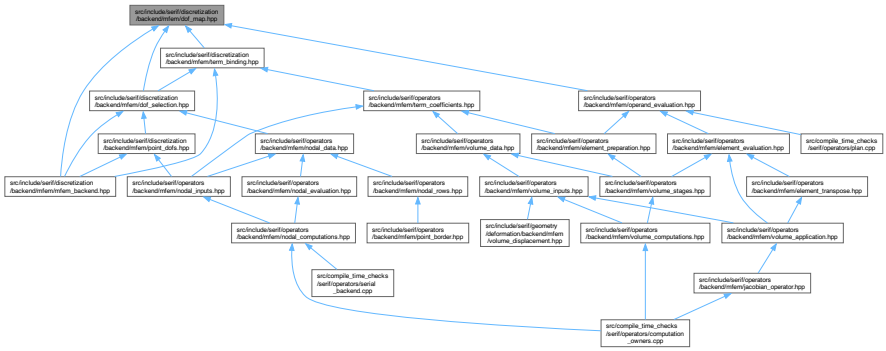
11.29 src/include/serif/discretization/backend/mfem/dof_map.hpp File Reference

This file contains the DofMap which allows for the full dof vector to be projected into a reduced space of dofs on supported elements for some field Generally, while much of the code here may be exercised, you will rarely see a DofMap be directly invoked. Rather, these are generally implicitly constructed through their child class term_binding.

#include <cmath>
#include <optional>
#include <stdexcept>
#include "mfem/fem/fespace.hpp"
#include "mfem/fem/pfespace.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for dof_map.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::discretization::backend::mfem::DofMap

  A mapping from a full MFEM finite-element space to a reduced set of DOFs that are supporter by the given element support. That is to say that if one has some set of dofs D (perhaps it looks like [0, 1, 2, 3, 4] as a hypothetical) and dofs 0, 2, and 4 are associated to domain A while 1 and 3 are associated to domain B. Then if one were to construct a DofMap with element support [1, 0, 1, 0, 1] (where 1 indicates that the element is supported and 0 indicates that it is not) then the reduced dof map would be [0, 2, 4] and the full to reduced map would be [-1, -1, 0, -1, 1].

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::backend
- namespace serif::discretization::backend::mfem

11.29.1 Detailed Description

This file contains the DofMap which allows for the full dof vector to be projected into a reduced space of dofs on supported elements for some field Generally, while much of the code here may be exercised, you will rarely see a DofMap be directly invoked. Rather, these are generally implicitly constructed through their child class term_binding.

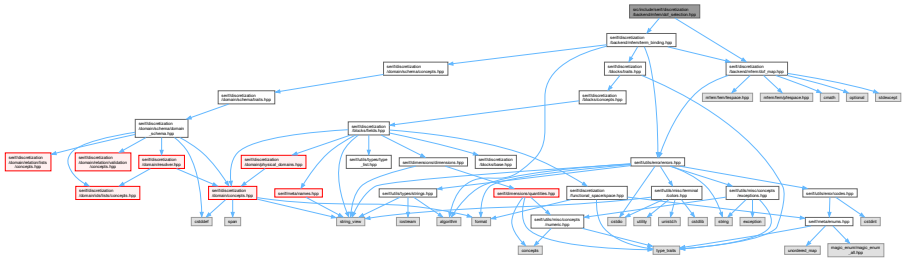
Author

Emily M. Boudreaux

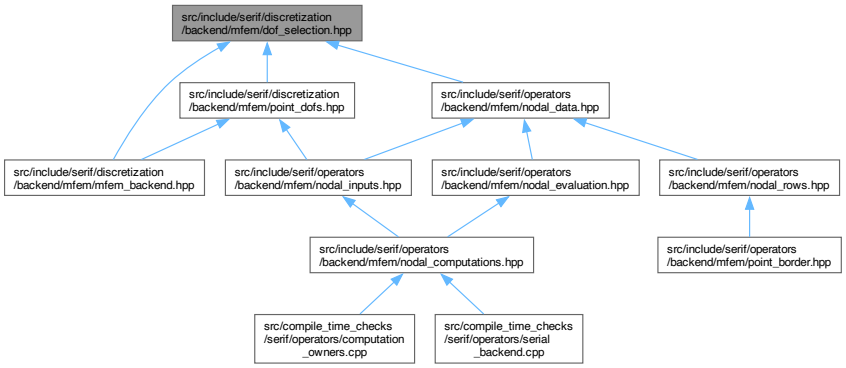
11.30 src/include/serif/discretization/backend/mfem/dof_selection.hpp File Reference

This file contains the DofSelection which allows for the selection of a subset of DOFs from a term binding, with boundary attribute information.

#include "serif/discretization/backend/mfem/term_binding.hpp"
#include "serif/discretization/backend/mfem/dof_map.hpp"
Include dependency graph for dof_selection.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::discretization::backend::mfem::DofSelection

  A selection of supported DOFs from a term binding, with boundary attribute information. The purpose here is to allow for the selection of a subset of DOFs from a term binding, such as those associated with a particular boundary, while still retaining the ability to validate the selection against the underlying mesh and term binding.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::backend
- namespace serif::discretization::backend::mfem

Functions

- template<domain::IsBoundary BoundaryT, blocks::SpatialTerm TermT, domain::schema::IsSchema SchemaT> DofSelection serif::discretization::backend::mfem::select_boundary_dofs (const TermBinding< TermT, SchemaT > &binding)

  Select DOFs associated with a particular boundary from a term binding.

11.30.1 Detailed Description

This file contains the DofSelection which allows for the selection of a subset of DOFs from a term binding, with boundary attribute information.

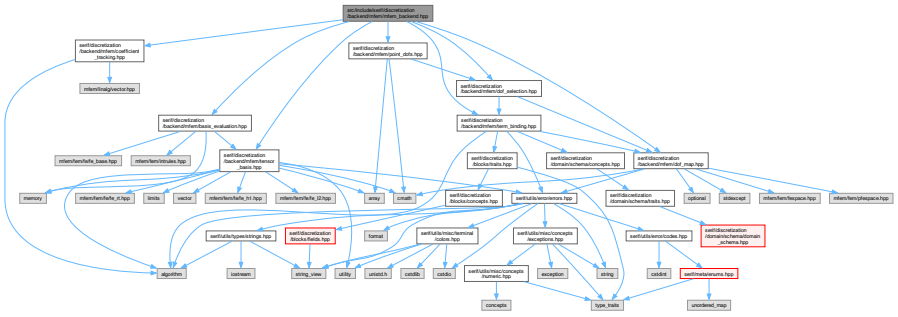
Author

Emily M. Boudreaux

11.31 src/include/serif/discretization/backend/mfem/mfem_backend.hpp File Reference

This is a utility file which allows one import to bring in all of the mfem discretization backend functionality. This file includes.

#include "serif/discretization/backend/mfem/basis_evaluation.hpp"
#include "serif/discretization/backend/mfem/coefficient_tracking.hpp"
#include "serif/discretization/backend/mfem/dof_map.hpp"
#include "serif/discretization/backend/mfem/dof_selection.hpp"
#include "serif/discretization/backend/mfem/point_dofs.hpp"
#include "serif/discretization/backend/mfem/tensor_basis.hpp"
#include "serif/discretization/backend/mfem/term_binding.hpp"
Include dependency graph for mfem_backend.hpp:



11.31.1 Detailed Description

This is a utility file which allows one import to bring in all of the mfem discretization backend functionality. This file includes.

Author

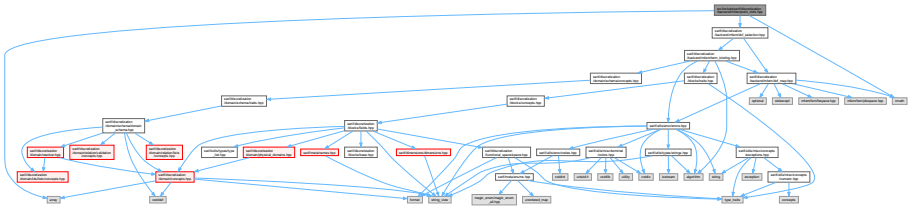
Emily M. Boudreaux

- dof_map.hpp
- dof_selection.hpp
- point_dofs.hpp
- tensor_basis.hpp
- term_binding.hpp

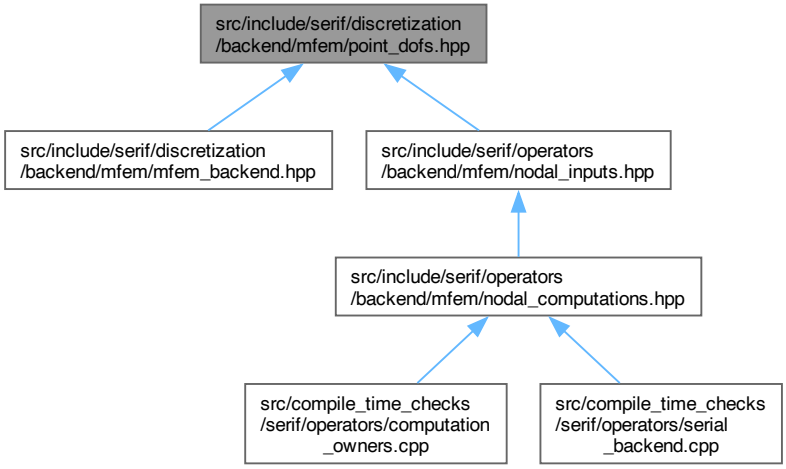
11.32 src/include/serif/discretization/backend/mfem/point_dofs.hpp File Reference

This file contains the implementation of function select_point_dofs. This function is used to select the components from a single H1 vertex in the fixed computational mesh. This is used to implement point constraints in the discretization (e.g. fixed center of mass)

#include <array>
#include <cmath>
#include "serif/discretization/backend/mfem/dof_selection.hpp"
Include dependency graph for point_dofs.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::backend
- namespace serif::discretization::backend::mfem

Functions

- template<blocks::SpatialTerm TermT, domain::schema::IsSchema SchemaT>
  requires (blocks::SpatialTermTraits<TermT>::family == functional_space::Family::h1)
  DofSelection serif::discretization::backend::mfem::select_point_dofs (const TermBinding< TermT, SchemaT > &binding, const std::array< double, 3 > &position={}, const double tolerance=1.0e-12)

  Select the dofs associated with a single point in the reference domain. This is used to implement point constraints in the discretization (e.g. fixed center of mass). Note that this function only works for H1 terms and will throw an error if used on a non-H1 term.

11.32.1 Detailed Description

This file contains the implementation of function select_point_dofs. This function is used to select the components from a single H1 vertex in the fixed computational mesh. This is used to implement point constraints in the discretization (e.g. fixed center of mass)

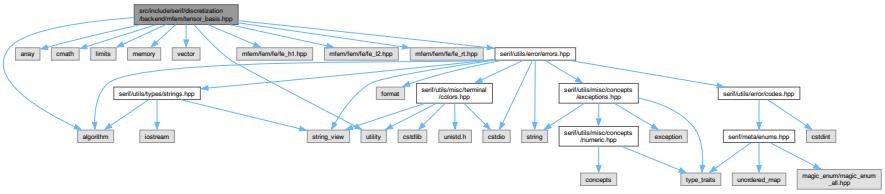
Author

Emily M. Boudreaux

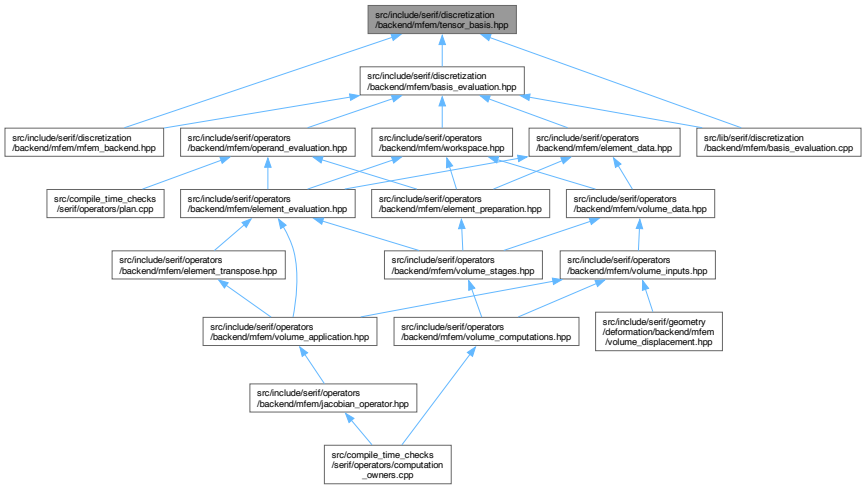
11.33 src/include/serif/discretization/backend/mfem/tensor_basis.hpp File Reference

Implementation of fast tensor evaluation functions. These do not contribute physics but are used to accelerate evaluations.

#include <algorithm>
#include <array>
#include <cmath>
#include <limits>
#include <memory>
#include <vector>
#include <utility>
#include "mfem/fem/fe/fe_h1.hpp"
#include "mfem/fem/fe/fe_l2.hpp"
#include "mfem/fem/fe/fe_rt.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for tensor_basis.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::backend::mfem::TensorWorkspace

  A workspace for tensor contractions to avoid repeated allocations and pack memory (TODO: does this actual pack memory in a useful manner when using mfem vectors? Probably something to look into).
- class serif::discretization::backend::mfem::TensorBasis

  A class that represents a tensor basis for a finite element. This class is used to evaluate the basis functions and their derivatives at a set of reference points.
- struct serif::discretization::backend::mfem::TensorBasis::Product

  A struct that represents a product of one-dimensional basis functions.
- class serif::discretization::backend::mfem::TensorBasisCache

  Share immutable tensor plans within one operator; copied rules are keyed by their contents.
- struct serif::discretization::backend::mfem::TensorBasisCache::Entry

  A struct that represents an entry in the TensorBasisCache. Each entry contains a non-owning pointer to a finite element, the reference operation, the number of copies, the points used to construct the TensorBasis, and a shared pointer to the TensorBasis itself.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::backend
- namespace serif::discretization::backend::mfem

Enumerations

- enum class serif::discretization::backend::mfem::ReferenceOperation { serif::discretization::backend::mfem::value , serif::discretization::backend::mfem::gradient , serif::discretization::backend::mfem::value_and_gradient , serif::discretization::backend::mfem::divergence }

  Options enum to control what is actually evaluated at the reference points.
- enum class serif::discretization::backend::mfem::ContractionMode { serif::discretization::backend::mfem::automatic , serif::discretization::backend::mfem::dense }

  Options enum to control how the tensor contractions are performed.

11.33.1 Detailed Description

Implementation of fast tensor evaluation functions. These do not contribute physics but are used to accelerate evaluations.

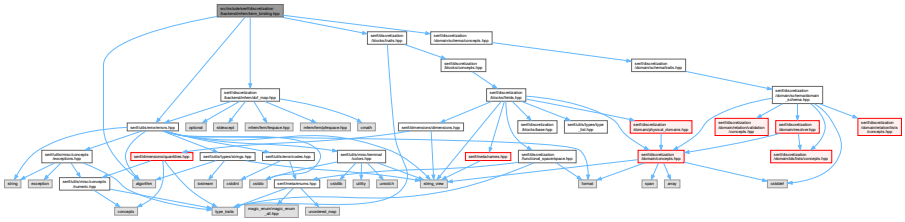
Author

Emily M. Boudreaux

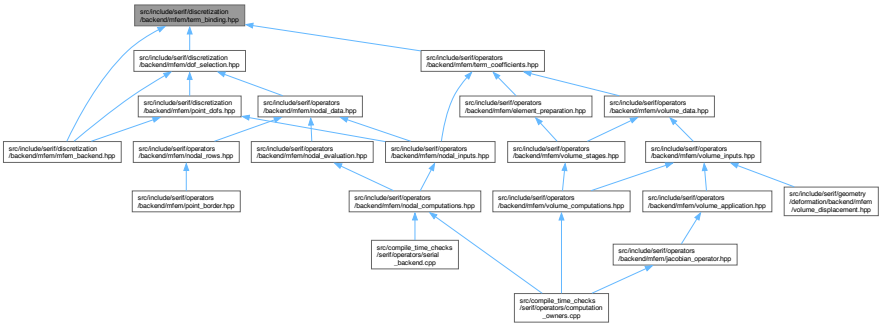
11.34 src/include/serif/discretization/backend/mfem/term_binding.hpp File Reference

A runtime binding between some term descriptor and its MFEM finite-element space. This is used to project down the compile time term descriptors into runtime reinterpretations which can then be used to extract out things like the list of dofs, the element support, and the mesh.

#include <algorithm>
#include "serif/discretization/blocks/traits.hpp"
#include "serif/discretization/domain/schema/concepts.hpp"
#include "serif/discretization/backend/mfem/dof_map.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for term_binding.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >

  A runtime binding between a term descriptor and its MFEM finite-element space. This is used to project down the compile time term descriptors into runtime reinterpretations which can then be used to extract out things like the list of dofs, the element support, and the mesh. Note that this is a child of DofMap, therefor all term bindings are also dof maps.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::backend
- namespace serif::discretization::backend::mfem

11.34.1 Detailed Description

A runtime binding between some term descriptor and its MFEM finite-element space. This is used to project down the compile time term descriptors into runtime reinterpretations which can then be used to extract out things like the list of dofs, the element support, and the mesh.

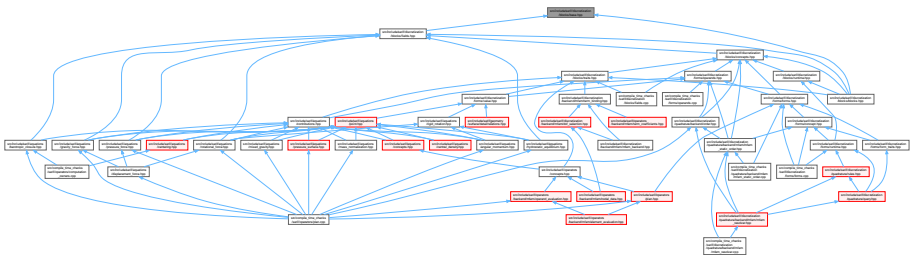
Author

Emily M. Boudreaux

11.35 src/include/serif/discretization/blocks/base.hpp File Reference

This file provides the base classes for residual and value blocks in the discretization framework.

This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::blocks::block

  Base class for all block types in the discretization framework.
- struct serif::discretization::blocks::residual_block_base

  Base class for residual blocks, derived from block.
- struct serif::discretization::blocks::value_block_base

  Base class for value blocks, derived from block.
- struct serif::discretization::blocks::generated_value_block< GeneratedValue >

  Final class for generated value blocks, derived from value_block_base. These are blocks that are generated at runtime.
- struct serif::discretization::blocks::generated_residual_block< GeneratedResidual >

  Final class for generated residual blocks, derived from residual_block_base. These are blocks that are generated at runtime.
- struct serif::discretization::blocks::residual_block< index_value >

  Final class for residual blocks with a fixed index, derived from residual_block_base.
- struct serif::discretization::blocks::value_block< index_value >

  Final class for value blocks with a fixed index, derived from value_block_base.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::blocks

Variables

- constexpr int serif::discretization::blocks::DYNAMIC_BLOCK_SIZE = -1

11.35.1 Detailed Description

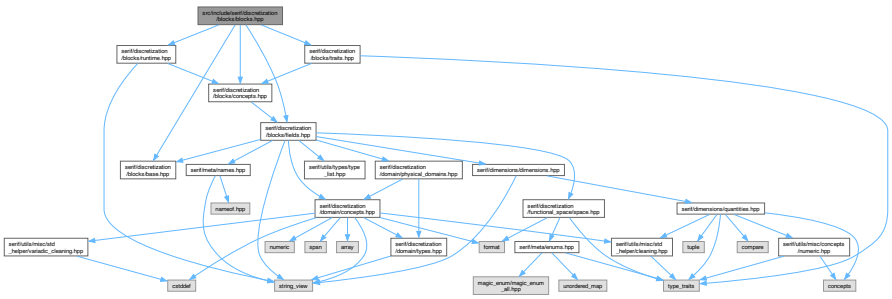
This file provides the base classes for residual and value blocks in the discretization framework.

Author

Emily M. Boudreaux

11.36 src/include/serif/discretization/blocks/blocks.hpp File Reference

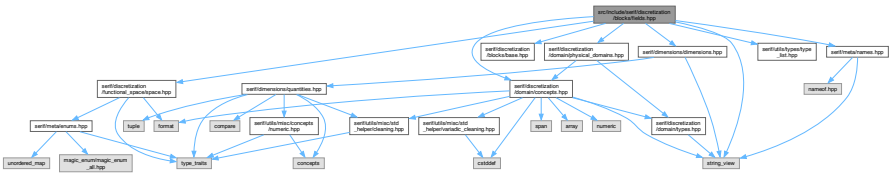
#include "serif/discretization/blocks/base.hpp"
#include "serif/discretization/blocks/concepts.hpp"
#include "serif/discretization/blocks/fields.hpp"
#include "serif/discretization/blocks/runtime.hpp"
#include "serif/discretization/blocks/traits.hpp"
Include dependency graph for blocks.hpp:



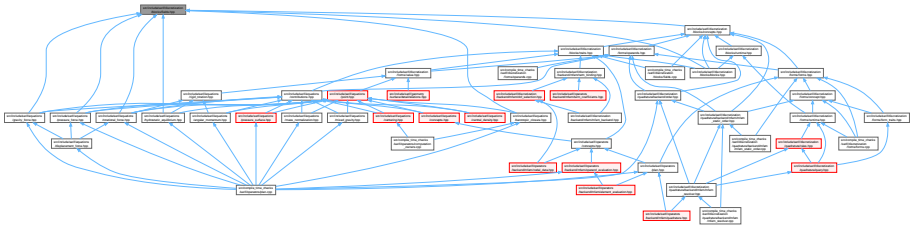
11.37 src/include/serif/discretization/blocks/fields.hpp File Reference

Define the fields and their associated terms for the discretization. This is a key file to understand if you are to understand SERiF's numerics. Each of these field types should contain all the information needed for the compiler to generate the correct.

#include <string_view>
#include "serif/dimensions/dimensions.hpp"
#include "serif/discretization/blocks/base.hpp"
#include "serif/discretization/functional_space/space.hpp"
#include "serif/discretization/domain/concepts.hpp"
#include "serif/discretization/domain/physical_domains.hpp"
#include "serif/utils/types/type_list.hpp"
#include "serif/meta/names.hpp"
Include dependency graph for fields.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::blocks::field

  The field struct is a tag type that is used to identify a field in the discretization. It is used to group together terms that belong to the same field. Each field should have a unique name and a list of terms that belong to it.
- struct serif::discretization::blocks::term_base

  The term_base struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Generally you should always use term rather than term_base as term associates to a particular field at compile time.
- struct serif::discretization::blocks::term< FieldT >

  The term struct is a tag type that is used to identify a term in the discretization. It is used to group together terms that belong to the same field. Each term should have a unique name and a list of blocks that belong to it. Note that generally you should always use term rather than term_base as term associates to a particular field at compile time.

- struct serif::discretization::blocks::global_scalar

  A global scalar is a term that has a single value over the entire domain (think angular momentum or total mass). Broadly we need to separate these from the other terms since they require different (often less enriched) information at compile time.
- struct serif::discretization::blocks::global_scalar::value

  The value block is a block that contains the value of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the actual value of the mass of the system.
- struct serif::discretization::blocks::global_scalar::residual

  The residual block is a block that contains the residual of the global scalar. It is a single value over the entire domain and is not associated with any particular functional space. For something like total mass this would be the residual of the mass of the system (i.e. the difference between the current mass and the target mass).
- struct serif::discretization::blocks::barotropic_constant

  The barotropic_constant field is a field that represents the barotropic constant of the system. It is a global scalar that is associated with the mass normalization term. This is used to enforce the mass normalization condition in the weak form when we are in the barotropic case. This field will not generalize as useful to the non-barotropic case. Note that we use compile time reflection to get the name of the field and the terms.
- struct serif::discretization::blocks::barotropic_constant::mass_normalization

  the mass_normalization term represents the mass normalization condition of the system. That is that it enforces that the total mass of the system is equal to the target mass in a weak sense. This is both a term and a global scalar.
- struct serif::discretization::blocks::rotation

  The rotational field, represented as the angular velocity term, given that we currently only model rigid rotation this is a global scalar. A future extension which models differential rotation will likely need to modify this to some vector field or other more complex representation of the rotation.
- struct serif::discretization::blocks::rotation::angular_velocity

  The angular_velocity term represents the angular velocity of the system. It is a global scalar that is associated with the rotation of the system. This is used to enforce the rotation condition in the weak form when we are in the rigid rotation case. Note that downstream the way that this currently works is to calculate the expected angular velocity given the current mass, geometry, density, and angular momentum and then prescribe, given those, what the angular velocity should be. Therefore, this does not enforce that the angular velocity must be constant over the entire solver run, but rather that the angular velocity is consistent with the current state of the system.
- struct serif::discretization::blocks::central_density

  field used to inject a border row used to enforce a central density constraint
- struct serif::discretization::blocks::central_density::border

  The border term represents the border row used to enforce a central density constraint. This is a global scalar as it is a single value over the entire domain.
- struct serif::discretization::blocks::density

  The density field used to represent the distribution of mass in the system. This is a scalar field that is associated with the mass density of the system. The density field is used to enforce the mass conservation condition in the weak form.
- struct serif::discretization::blocks::density::mass

  The mass density term for the system. Note that the naming here could be a bit confusing as FEM also uses the term ”mass” to represent the canonical mass matrix. This is a more standard physics reading of what is going on here. We define that this space is a second order scalar L2 space. This allows for the possibility of discontinuous density fields.
- struct serif::discretization::blocks::density::mass::value

  The value block is a block that contains the value of the mass density term. It is a scalar field that is associated with the mass density of the system.
- struct serif::discretization::blocks::density::mass::residual

The residual block is a block that contains the residual of the mass density term. It is a scalar field that is associated with the mass density of the system.

- struct serif::discretization::blocks::displacement

  The displacement field is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain.

- struct serif::discretization::blocks::displacement::vector

  the displacement vector term is a 3 vector H1 field (C0 continuous) that is used to represent continuous nodal displacement over the entire domain. This is a vector field that is associated with the mapping of the domain from the reference to physical spaces. E.g. if there is some domain O_ref which n nodal positions x^i_ref then there is some other domain O_phys with n nodel positions such that x^i_phys = x^i_ref + d^i where d^i is the displacement vector at node i.

- struct serif::discretization::blocks::displacement::vector::value

  The value block is a block that contains the value of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces.

- struct serif::discretization::blocks::displacement::vector::residual

  The residual block is a block that contains the residual of the displacement vector term. It is a vector field that is associated with the mapping of the domain from the reference to physical spaces.

- struct serif::discretization::blocks::surface_deformation

  The surface corollary field to the full displacement field. This is in some ways more general than the others in so far as it is not pointwise defined over the domain. Rather, it is some general field of numbers, divorced from the underlying discrimination, that is used to represent the surface deformation of the system. This may be used, for example to represent spherical harmonic coefficients of the surface deformation. For more information see the surface and deformation systems.

- struct serif::discretization::blocks::surface_deformation::coefficients

  The coefficients term represents some set of coefficients associated with whatever surface representation is being used to represent the surface deformation. Note how we do not have a functional space, support, or a quantity. This is also where we associate the values with.

- struct serif::discretization::blocks::surface_deformation::coefficients::value

  The actual values of the coefficients for whatever surface representation is being used.

- struct serif::discretization::blocks::surface_deformation::shape_equilibrium

  The shape equilibrium term represents the shape equilibrium condition of the system. This is a term that is associated with the surface deformation field and is used to enforce the shape equilibrium condition in the weak form. This is where we associate the residuals with. Note how we do not have a functional space, support, or a quantity. This is because the shape equilibrium condition is a global condition that is not associated with any particular functional space, support, or quantity.

- struct serif::discretization::blocks::surface_deformation::shape_equilibrium::residual

  The residuals of the shape equilibrium condition.

- struct serif::discretization::blocks::gravity

  The gravity field is represented as a mixed system, in many ways this field is the motivation for the entire design of this file. Since it is a mixed system we have two separate terms that make it up, the potential and the gradient. These are not derived from one and other, rather they are weakly enforced to agree. This buys us an additional order of convergence in the potential which in tern allows us to use a lower resolution mesh.

- struct serif::discretization::blocks::gravity::gradient

  The gravitational gradient term is a 2nd order Hdiv vector field that is used to represent the negative gravitational acceleration of the system. Note that when I started developing this code I made the frankly wacky choice to let this be the true gradient of the potential, which is the negative acceleration. Broadly this may make sense mathematically; however, it means that developers must be extra careful to remember the sign convention.

- struct serif::discretization::blocks::gravity::gradient::value

  The value block is a block that contains the value of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system.

- struct serif::discretization::blocks::gravity::gradient::residual

The residual block is a block that contains the residual of the gravitational gradient term. It is a vector field that is associated with the negative gravitational acceleration of the system.

- struct serif::discretization::blocks::gravity::potential

  The gravitational potential term is a 2nd order L2 scalar field that is used to represent the gravitational potential of the system. Note that this is not exactly the same as the if you were to take the scalar potential of the gravitational gradient term. These are solved for by the code separately and weakly enforced to agree. Generally if you need the most accurate gradient you should use the gradient term and if you need the most accurate potential you should use this term (this is because differentiating this term loses an order of accuracy in the differentiation).
- struct serif::discretization::blocks::gravity::potential::value

  The value block is a block that contains the value of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system.
- struct serif::discretization::blocks::gravity::potential::residual

  The residual block is a block that contains the residual of the gravitational potential term. It is a scalar field that is associated with the gravitational potential of the system.
- struct serif::discretization::blocks::enthalpy

  The enthalpy field is a scalar field is a C0 continuous scalar field. Note that this field likely will not survive the transition to the more general non-barotropic case. In the barotropic case the gradient of enthalpy is parallel to the gradient of the pressure. This make enthalpy a particularly useful quantity as we can represent the 0 pressure surface condition as the 0 enthalpy condition and we can break down mass constraint into a game of optimizing the Bernoulli constant.
- struct serif::discretization::blocks::enthalpy::specific

  the specific enthalpy is the term we specifically work with rather than the total enthalpy of dof. In fact heuristically I would say that generally we should always work in specific quantities rather than total quantities as there is not a well defined amount of material at some dof.
- struct serif::discretization::blocks::enthalpy::specific::value

  The value block is a block that contains the value of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system.
- struct serif::discretization::blocks::enthalpy::specific::residual

  The residual block is a block that contains the residual of the specific enthalpy term. It is a scalar field that is associated with the specific enthalpy of the system.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::blocks

11.37.1 Detailed Description

Define the fields and their associated terms for the discretization. This is a key file to understand if you are to understand SERiF's numerics. Each of these field types should contain all the information needed for the compiler to generate the correct.

Author

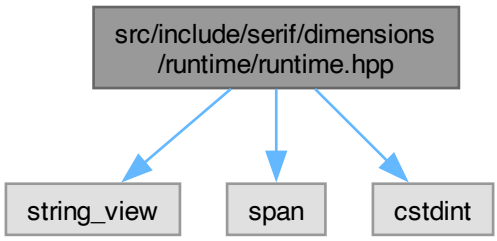
Emily M. Boudreaux

- blocks
- functional spaces
- support domains
- quantities
- symbols
- and any other information needed to generate the correct weak form for the field Broadly you can think of these as where we define the ”physics” that we want to solve for or the unknowns of the system. If you find yourself adding any sort of unknown to the system and you are not adding it to this file you should step back and reconsider\ if that belongs here instead.

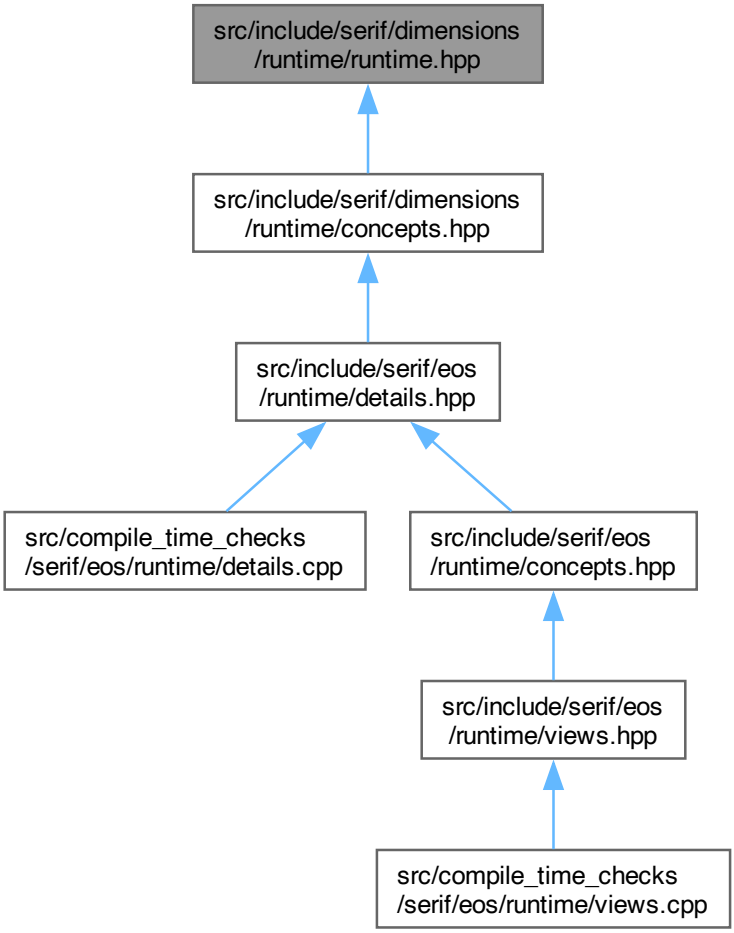
11.38 src/include/serif/dimensions/runtime/runtime.hpp File Reference

runtime usable types representing the compile time dimension type system. Note that generally this is not a dimensional analysis library. Though perhaps in the future it will grow to something like that. Rather, this system is intended to ensure that developers are explicit which what numeric values mean. There is more detail on this in the file comment for dimensions.hpp and quantities.hpp

#include <string_view>
#include
#include <cstdint>
Include dependency graph for runtime.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::dimensions::runtime::ThermodynamicQuantityID

  Runtime resolvable ID for a thermodynamic quantity. Used in conjunction with concepts to resolve compile time thermodynamic types to runtime values.
- struct serif::dimensions::runtime::RuntimeQuantityValue

  Runtime resolved type for a quantity value. This holds both an ID and a value. This is the runtime equivalent to a QuantityValue. Note that where these differ is that QuantityValue is built with template types. This allows the compiler to resolve it to a plain numeric type and still do optimizations That is generally not the case with RuntimeQuantityValue.
- struct serif::dimensions::runtime::RuntimeRelationDescriptor

Namespaces

- namespace serif
- namespace serif::dimensions
- namespace serif::dimensions::runtime

11.38.1 Detailed Description

runtime usable types representing the compile time dimension type system. Note that generally this is not a dimensional analysis library. Though perhaps in the future it will grow to something like that. Rather, this system is intended to ensure that developers are explicit which what numeric values mean. There is more detail on this in the file comment for dimensions.hpp and quantities.hpp

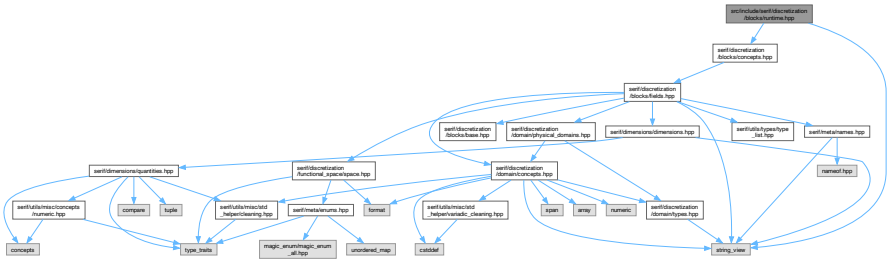
Author

Emily M. Boudreaux

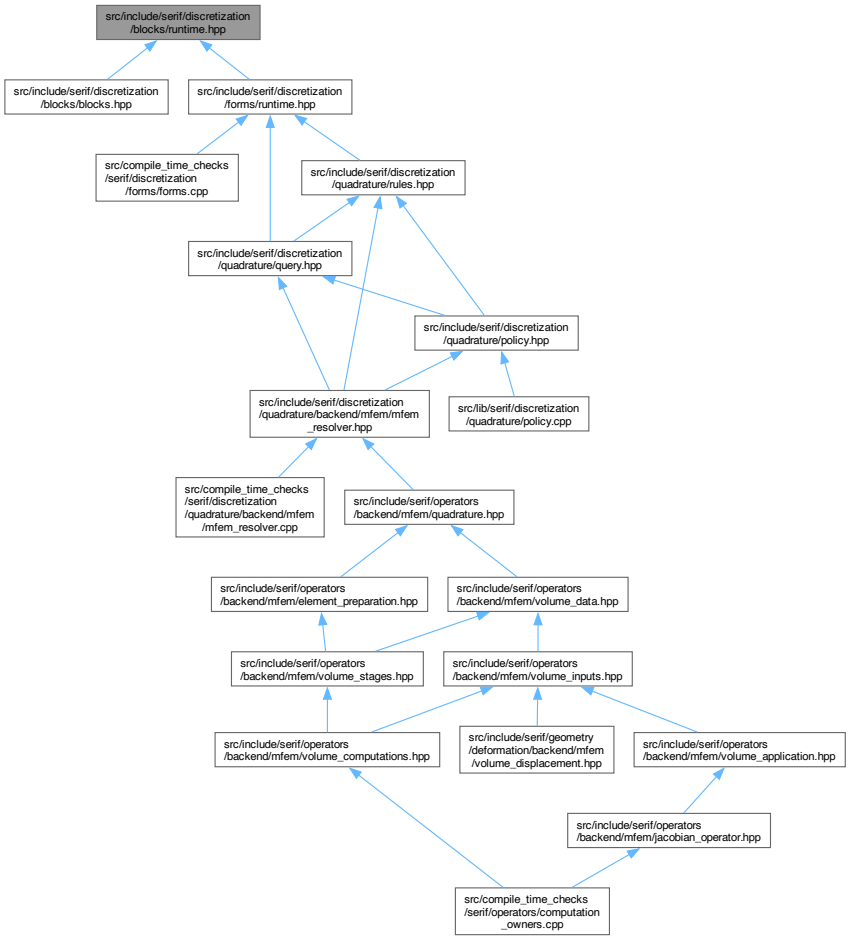
11.39 src/include/serif/discretization/blocks/runtime.hpp File Reference

Runtime descriptors for fields and terms, used to project compile time types into runtime values.

#include <string_view>
#include "serif/discretization/blocks/concepts.hpp"
Include dependency graph for runtime.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::blocks::FieldDescriptor

  Runtime descriptor for a field, containing its name. Two FieldDescriptors are equal if they have identical names.
- struct serif::discretization::blocks::TermDescriptor

  Runtime descriptor for a term, containing its associated field and name. Two TermDescriptors are equal if they have identical fields and names.
- struct std::formatter< serif::discretization::blocks::FieldDescriptor >

  Specialization of std::formatter for FieldDescriptor and TermDescriptor to enable formatting with std::format.
- struct std::formatter< serif::discretization::blocks::TermDescriptor >

  Specialization of std::formatter for TermDescriptor to enable formatting with std::format.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::blocks

Variables

- template<Field FieldT>
  constexpr FieldDescriptor serif::discretization::blocks::field_descriptor {.name = FieldT::name}

  Create a FieldDescriptor for a given field type.
- template<Term TermT>
  constexpr TermDescriptor serif::discretization::blocks::term_descriptor

  Create a TermDescriptor for a given term type.

11.39.1 Detailed Description

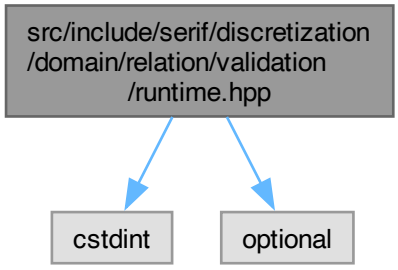
Runtime descriptors for fields and terms, used to project compile time types into runtime values.

Author

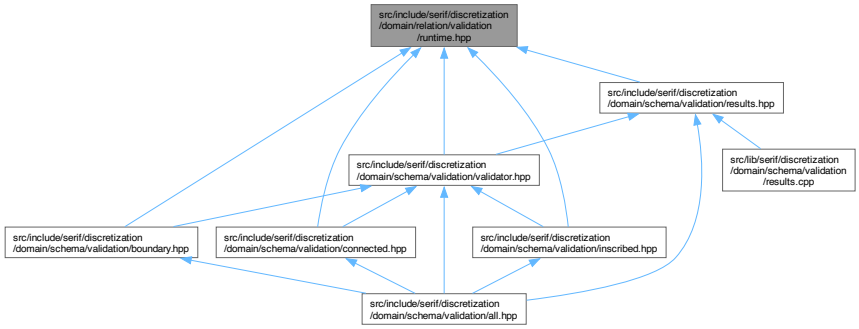
Emily M. Boudreaux

11.40 src/include/serif/discretization/domain/relation/validation/runtime.hpp File Reference

#include <cstdint>
#include <optional>
Include dependency graph for runtime.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::relation::validation::RelationValidationResult
- struct serif::discretization::domain::relation::validation::RelationValidationResult::InscribedDiagnostics
- struct serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics
- struct serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::relation
- namespace serif::discretization::domain::relation::validation

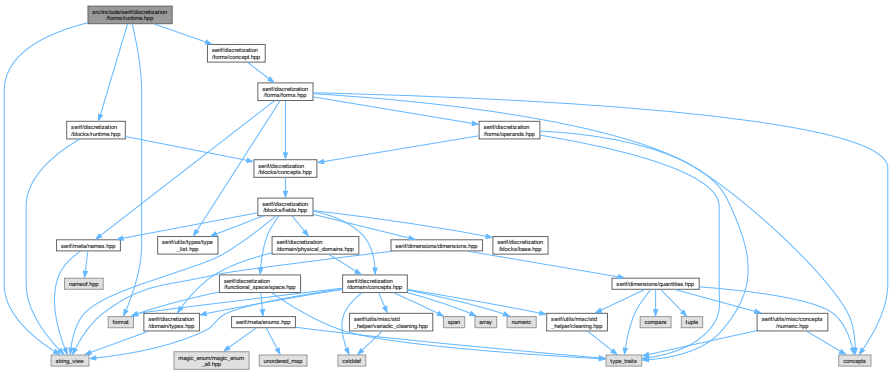
Enumerations

- enum class serif::discretization::domain::relation::validation::RelationValidationFailure : std::uint8_t { serif::discretization::domain::relation::validation::None , serif::discretization::domain::relation::validation::DomainAbsent , serif::discretization::domain::relation::validation::DomainDisconnected , serif::discretization::domain::relation::validation::OuterDomainAbsent , serif::discretization::domain::relation::validation::InnerDomainTouchesMeshBoundary , serif::discretization::domain::relation::validation::DomainBoundaryAbsent , serif::discretization::domain::relation::validation::DomainBoundaryTaggedFaceTouchesUnexpectedDomain , serif::discretization::domain::relation::validation::DomainBoundaryExpectedFaceIsUntagged , serif::discretization::domain::relation::validation::DomainBoundaryExpectedFaceHasWrongID }

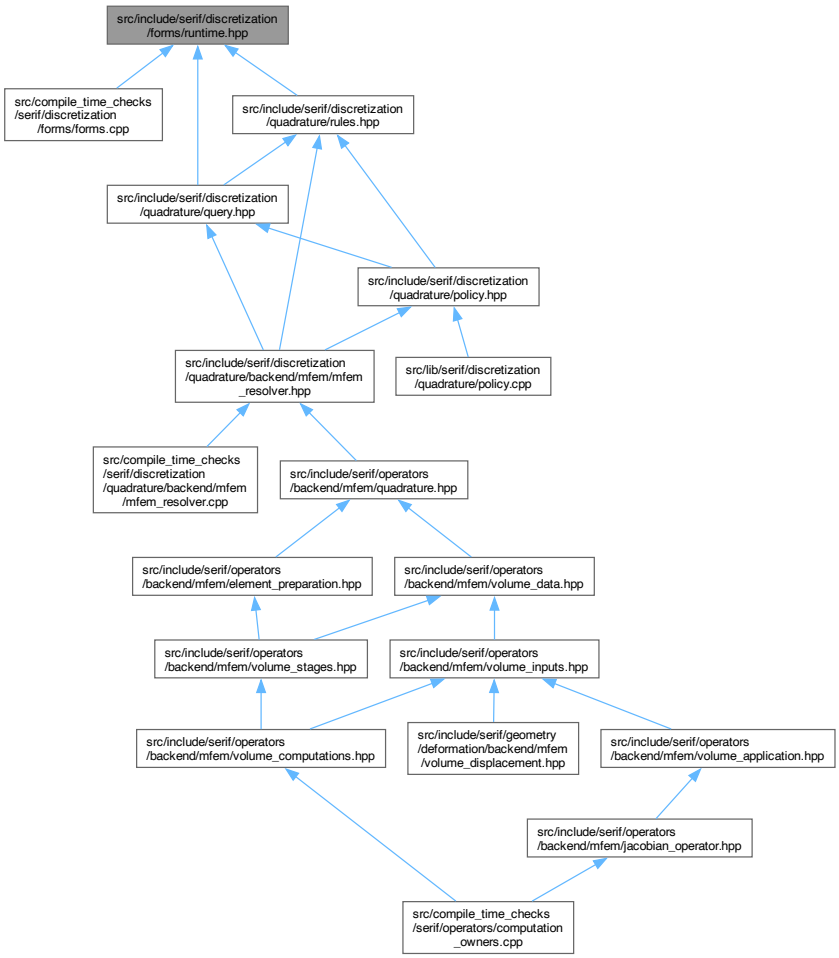
11.41 src/include/serif/discretization/forms/runtime.hpp File Reference

file defining some utilities to project form information from compile time into runtime

#include <string_view>
#include <format>
#include "serif/discretization/blocks/runtime.hpp"
#include "serif/discretization/forms/concept.hpp"
Include dependency graph for runtime.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::forms::FormDescriptor

  Runtime form descriptor including runtime term information. Used to represented the compile-time form type at runtime.
- struct std::formatter< serif::discretization::forms::FormDescriptor >

  Overload to allow println for FormDescriptors.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::forms

Variables

- template<Form FormT>
  constexpr FormDescriptor serif::discretization::forms::form_descriptor

  Specialization of the FormDescriptor class at compile time for some particular compile-time form. This allows us to generate constexpr instances of this struct at compile time and usable at runtime.

11.41.1 Detailed Description

file defining some utilities to project form information from compile time into runtime

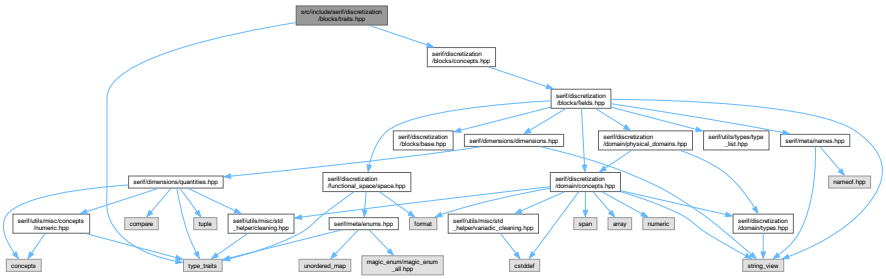
Author

Emily M. Boudreaux

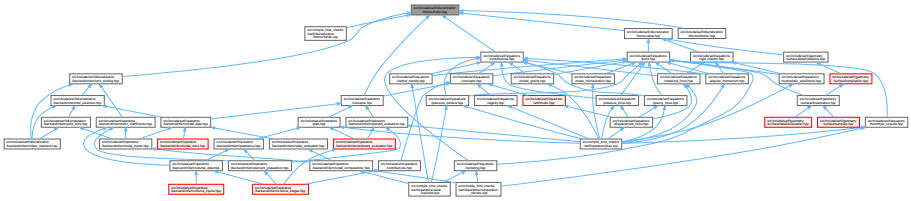
11.42 src/include/serif/discretization/blocks/traits.hpp File Reference

Compile-time traits for fields and terms, used to extract types and properties from template parameters.

#include <type_traits>
#include "serif/discretization/blocks/concepts.hpp"
Include dependency graph for traits.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::blocks::SpatialTermTraits< TermT >

  Helper struct representing the traits of some spatial term type, including its functional space, support, family, order, and rank.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::blocks

Typedefs

- template<Term TermT>
  using serif::discretization::blocks::FieldOf = std::remove_cvref_t<TermT>::field_type

  Extract the field type associated with a given term type. Will have a type of whatever TermT::field_type is.
- template<HasFunctionalSpace TermT>
  using serif::discretization::blocks::FunctionalSpaceOf = std::remove_cvref_t<TermT>::FunctionalSpace

  Extract the functional space type associated with a given term type. Will have a type of whatever TermT::FunctionalSpace is.
- template<HasSupport TermT>
  using serif::discretization::blocks::SupportOf = std::remove_cvref_t<TermT>::Support

  Extract the support type associated with a given term type. Will have a type of whatever TermT::Support is.
- template<HasQuantity TermT>
  using serif::discretization::blocks::QuantityOf = std::remove_cvref_t<TermT>::Quantity

  Extract the quantity type associated with a given term type. Will have a type of whatever TermT::Quantity is.
- template<typename FieldT>
  using serif::discretization::blocks::TermsOf = FieldT::Terms

  Extract the terms associated with a given field type. Will have a type of whatever FieldT::Terms is.

11.42.1 Detailed Description

Compile-time traits for fields and terms, used to extract types and properties from template parameters.

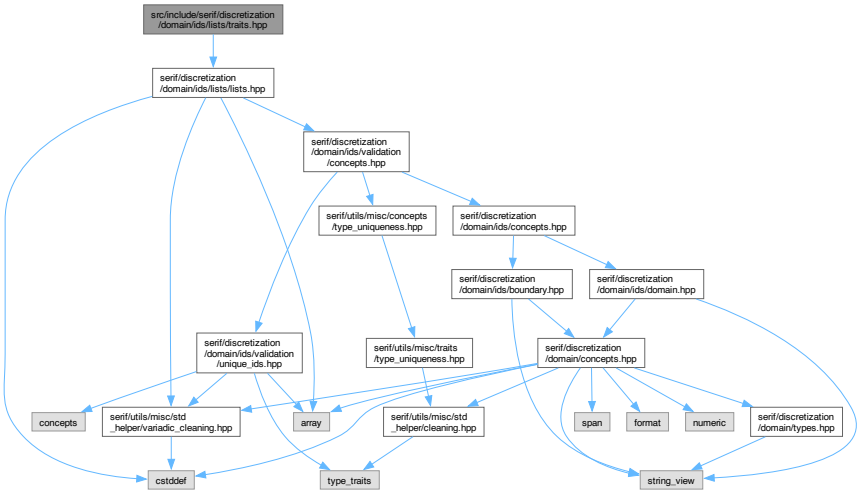
Author

Emily M. Boudreaux

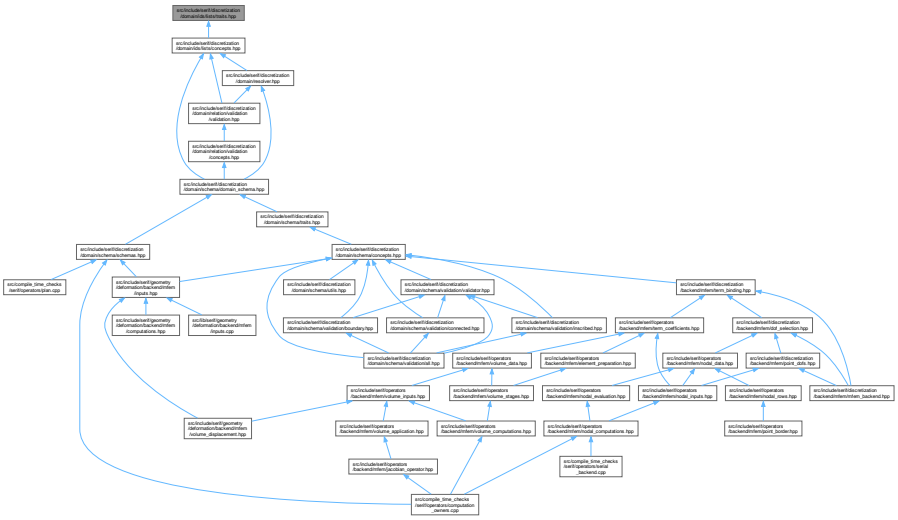
11.43 src/include/serif/discretization/domain/ids/lists/traits.hpp File Reference

Compile-time traits for domain ID lists, used to extract types and properties from template parameters.

#include "serif/discretization/domain/ids/lists/lists.hpp"
Include dependency graph for traits.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif

- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::ids
- namespace serif::discretization::domain::ids::lists

Variables

- template<typename T>
  constexpr bool serif::discretization::domain::ids::lists::is_domain_id_list_v = false

  Trait to determine if a type is a domain ID list. Defaults to false for all types.
- template<IsDomainID... DomainIDTs>
  constexpr bool serif::discretization::domain::ids::lists::is_domain_id_list_v< DomainIDList< DomainIDTs... > > = true

  Trait to determine if a type is a boundary ID list. Specialization which is true for valid BoundaryIDList types.
- template<typename T>
  constexpr bool serif::discretization::domain::ids::lists::is_boundary_id_list_v = false

  Trait to determine if a type is a boundary ID list. Defaults to false for all types.
- template<IsBoundaryID... BoundaryIDTs>
  constexpr bool serif::discretization::domain::ids::lists::is_boundary_id_list_v< BoundaryIDList< BoundaryIDTs... > > = true

  Trait to determine if a type is a boundary ID list. Specialization which is true for valid BoundaryIDList types.

11.43.1 Detailed Description

Compile-time traits for domain ID lists, used to extract types and properties from template parameters.

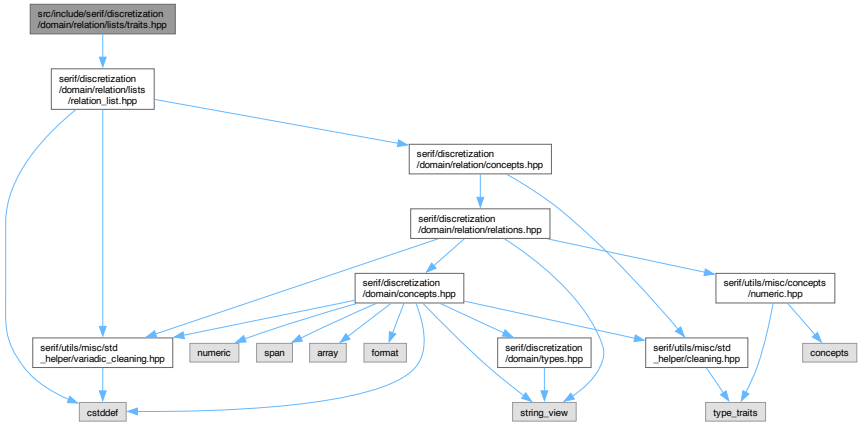
Author

Emily M. Boudreaux

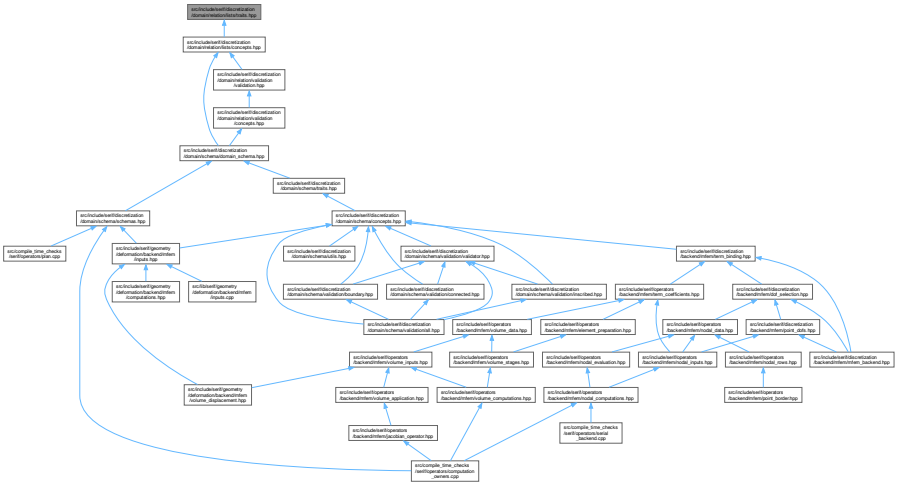
11.44 src/include/serif/discretization/domain/relation/lists/traits.hpp File Reference

#include "serif/discretization/domain/relation/lists/relation_list.hpp"

Include dependency graph for traits.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

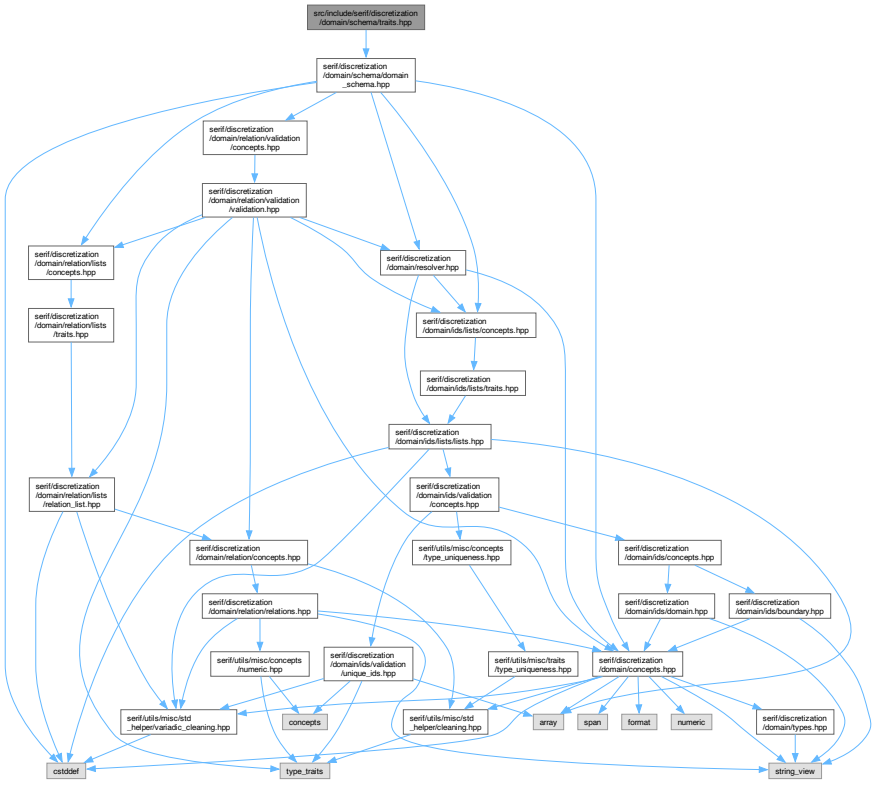
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::relation
- namespace serif::discretization::domain::relation::lists

Variables

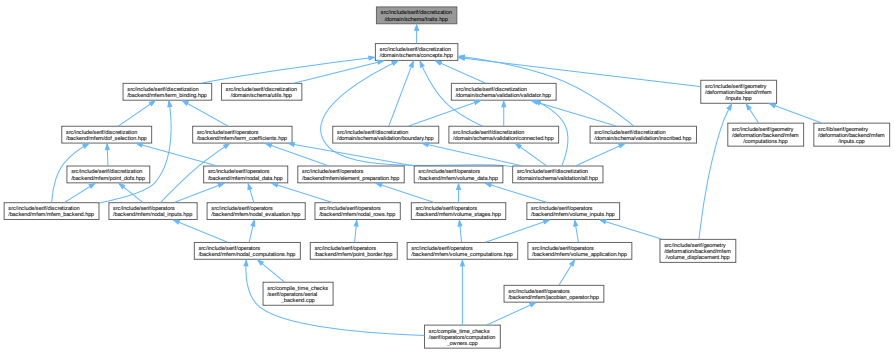
- template<typename T>
  constexpr bool serif::discretization::domain::relation::lists::is_relation_list_v = false
- template<IsRelation... RelationTs>
  constexpr bool serif::discretization::domain::relation::lists::is_relation_list_v< RelationList< RelationTs... > >
  = true

11.45 src/include/serif/discretization/domain/schema/traits.hpp File Reference

#include "serif/discretization/domain/schema/domain_schema.hpp"
Include dependency graph for traits.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

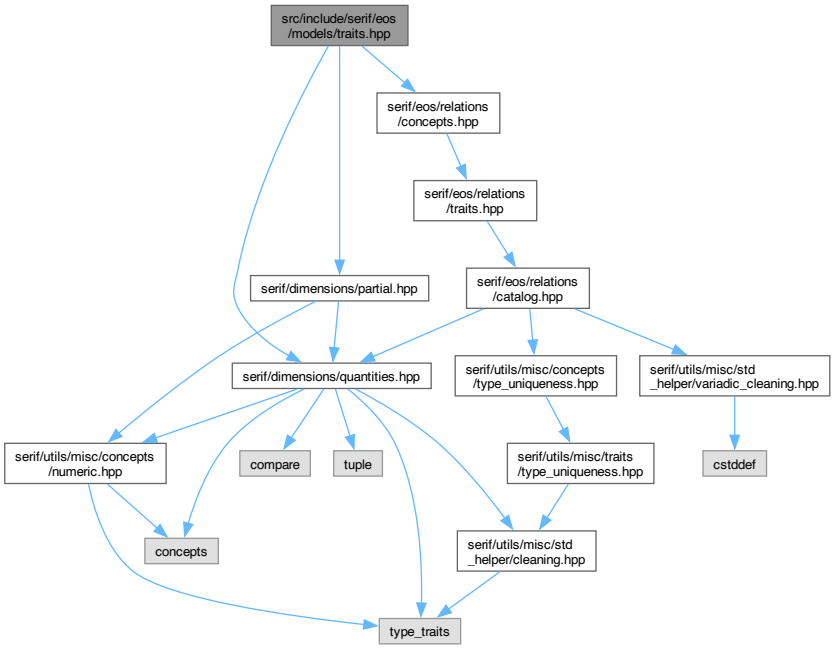
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema

Variables

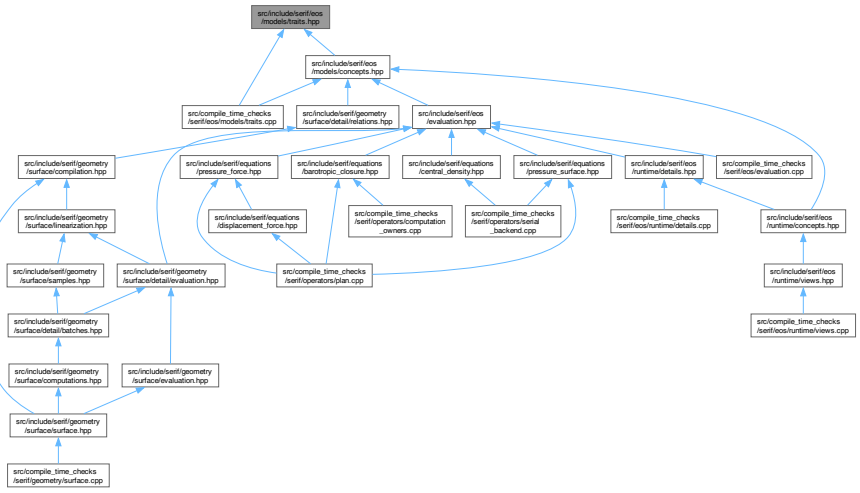
- template<typename S>
  constexpr bool serif::discretization::domain::schema::is_schema_v = false
- template<IsDomainIDList DomainIDs, IsBoundaryIDList BoundaryIDs, IsRelationList Relations>
  constexpr bool serif::discretization::domain::schema::is_schema_v< DomainSchema< DomainIDs, BoundaryIDs, Relat
  = true

11.46 src/include/serif/eos/models/traits.hpp File Reference

#include ”serif/dimensions/quantities.hpp”
#include ”serif/dimensions/partial.hpp”
#include ”serif/eos/relations/concepts.hpp”
Include dependency graph for traits.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::eos::models::ImplementsRelation_t< EOS, RelationT >
- struct serif::eos::models::ImplementsRelationCatalog_t< EOS, Output, Inputs >
- struct serif::eos::models::ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > >
- struct serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, typename >
- struct serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std::remove_cvref_t< E
- struct serif::eos::models::ImplementsPartialDerivative_t< EOS, RelationT, InputQuantity >

Namespaces

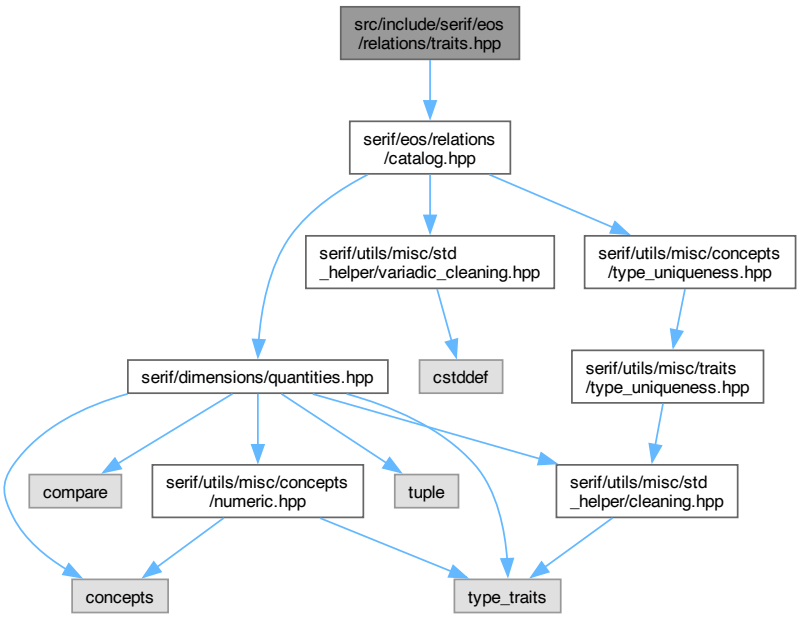
- namespace serif
- namespace serif::eos
- namespace serif::eos::models

Variables

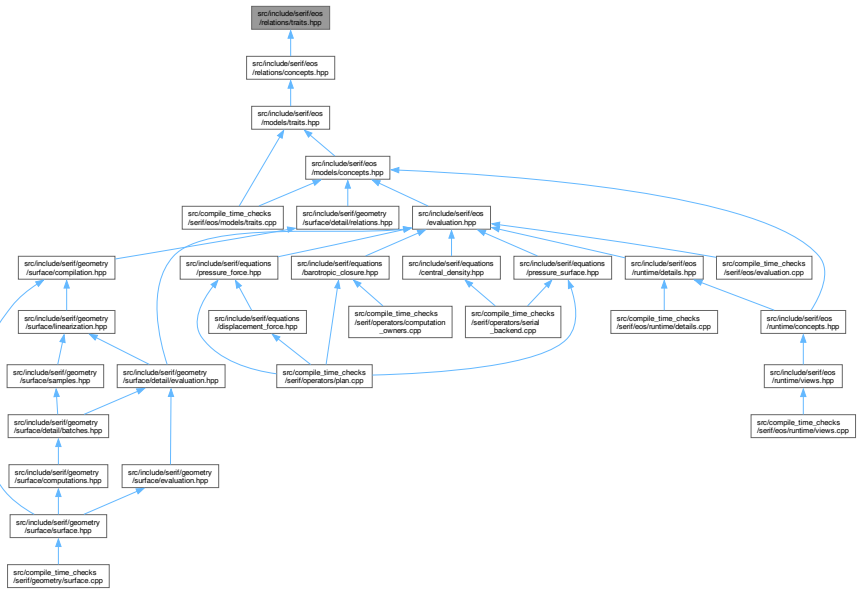
- template<typename EOSCandidate>
  constexpr bool serif::eos::models::is_equation_of_state_model_v = IsEquationOfStateModel_t<std↩
  ::remove_cvref_t<EOSCandidate>>::value
- template<typename EOS, typename RelationT>
  constexpr bool serif::eos::models::supports_relation_v = is_equation_of_state_model_v<EOS> &&
  ImplementsRelation_t<std::remove_cvref_t<EOS>, std::remove_cvref_t<RelationT>>::value
- template<typename EOS, typename RelationT, typename InputQuantity>
  constexpr bool serif::eos::models::supports_partial_derivative_v = is_equation_of_state_model_v<EOS>
  && ImplementsPartialDerivative_t<std::remove_cvref_t<EOS>, std::remove_cvref_t<RelationT>,
  InputQuantity>::value

11.47 src/include/serif/eos/relations/traits.hpp File Reference

#include "serif/eos/relations/catalog.hpp"
Include dependency graph for traits.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::eos::relations::CatalogContainsRelation< Catalog, RelationT >
- struct serif::eos::relations::CatalogContainsRelation< RelationCatalog< Relations... >, RelationT >
- struct serif::eos::relations::RelationContainsInput< RelationT, Quantity >
- struct serif::eos::relations::RelationContainsInput< Relation< Output, Inputs... >, Quantity >
- struct serif::eos::relations::IsThermodynamicRelation< RelationCandidate >
- struct serif::eos::relations::IsThermodynamicRelation< Relation< Output, Inputs... > >
- struct serif::eos::relations::IsValidRelationCatalog< CandidateCatalog >
- struct serif::eos::relations::IsValidRelationCatalog< RelationCatalog< Relations... > >

Namespaces

- namespace serif
- namespace serif::eos
- namespace serif::eos::relations

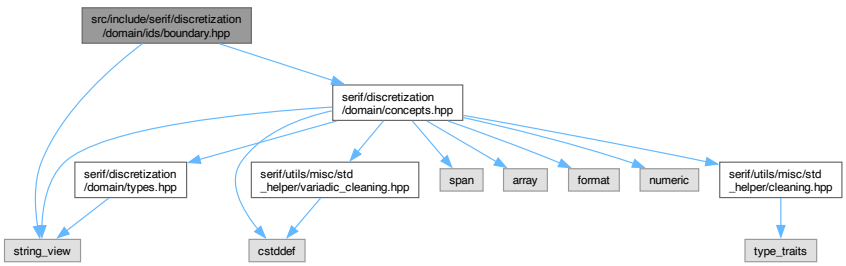
Variables

- template<typename Catalog, typename RelationT>
  constexpr bool serif::eos::relations::relation_catalog_contains_relation_v = CatalogContainsRelation<std::remove_cv_t<Catalog>, std::remove_cv_t<RelationT>>::value
- template<typename RelationT, typename Quantity>
  constexpr bool serif::eos::relations::relation_contains_input_v = RelationContainsInput<std::remove_cv_t<RelationT>, std::remove_cv_t<Quantity>>::value
- template<typename RelationCandidate>
  constexpr bool serif::eos::relations::is_thermodynamic_relation_v = IsThermodynamicRelation<RelationCandidate>::value
- template<typename CatalogCandidate>
  constexpr bool serif::eos::relations::is_valid_relation_catalog_v = IsValidRelationCatalog<CatalogCandidate>::value

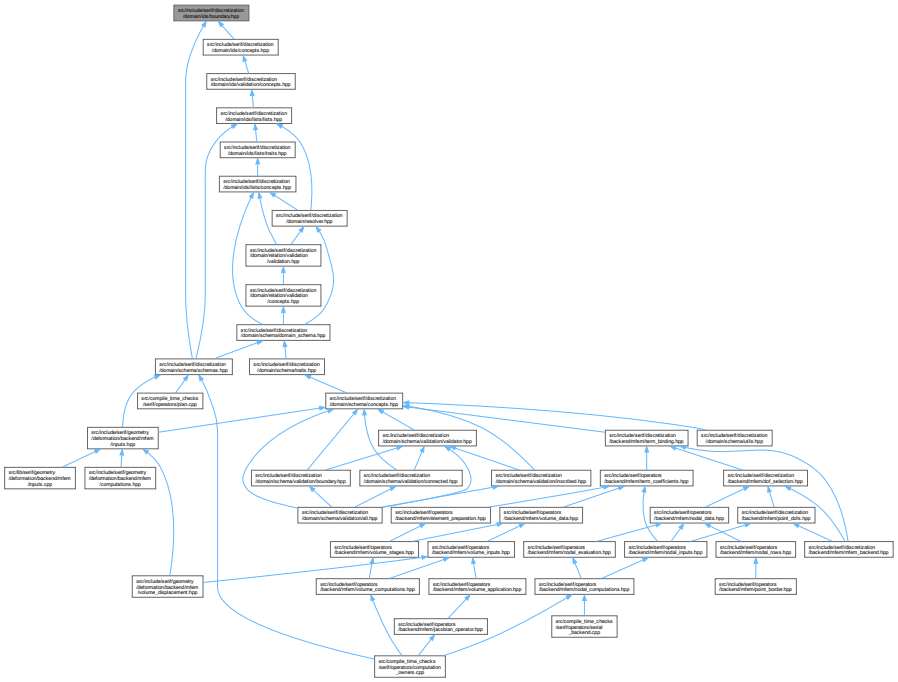
11.48 src/include/serif/discretization/domain/ids/boundary.hpp File Reference

This file defines the BoundaryID type, which is used to bind a semantic surface (a Boundary) to the integer boundary attribute MFEM stores in the mesh file.

#include <string_view>
#include "serif/discretization/domain/concepts.hpp"
Include dependency graph for boundary.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >

  The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation.
- struct serif::discretization::domain::ids::BoundaryDescriptor

  The BoundaryDescriptor struct is a runtime representation of a boundary, containing its name and integer attribute ID. This is used to project compile-time BoundaryID types into runtime values.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::ids

11.48.1 Detailed Description

This file defines the BoundaryID type, which is used to bind a semantic surface (a Boundary) to the integer boundary attribute MFEM stores in the mesh file.

Author

Emily M. Boudreaux

For example, the following code defines a BoundaryID that associates the StellarSurfaceBoundary with the integer attribute 1 in the mesh file:

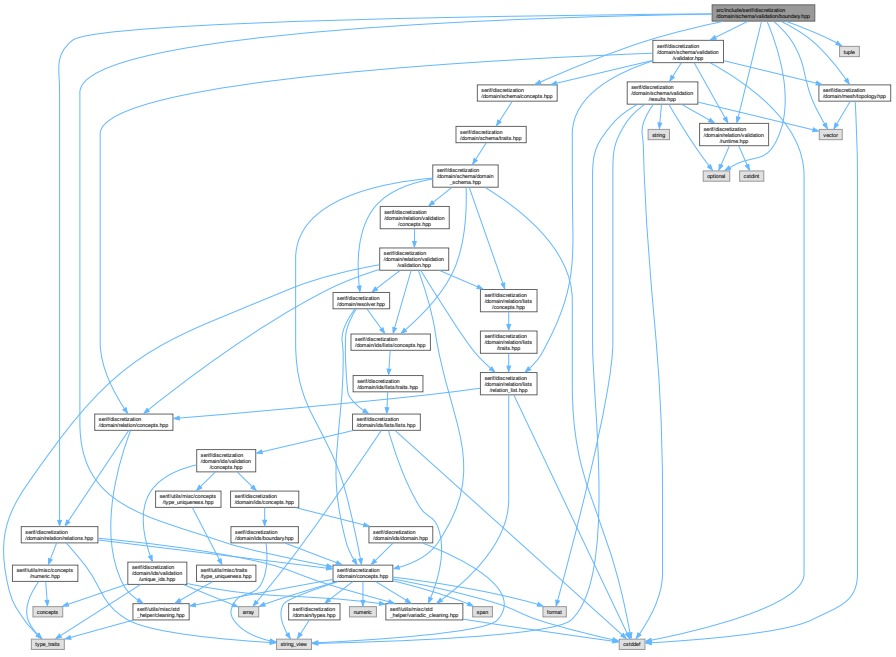
```
BoundaryID<StellarSurfaceBoundary, 1> stellarSurfaceBoundaryID;
```

Read this as ”boundary elements with attribute 1 are the stellar surface”.

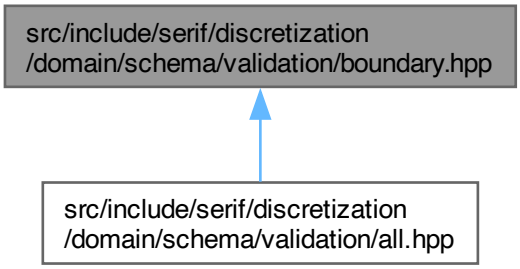
11.49 src/include/serif/discretization/domain/schema/validation/boundary.hpp File Reference

```
#include <optional>
#include <tuple>
#include <vector>
#include ”serif/discretization/domain/concepts.hpp”
#include ”serif/discretization/domain/mesh/topology.hpp”
#include ”serif/discretization/domain/relation/relations.hpp”
#include ”serif/discretization/domain/relation/validation/runtime.hpp”
#include ”serif/discretization/domain/schema/concepts.hpp”
#include ”serif/discretization/domain/schema/validation/validator.hpp”
```

Include dependency graph for boundary.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::schema::validation::RelationValidator< DomainBoundary< BoundaryT, DomainTs.
- struct serif::discretization::domain::schema::validation::DomainBoundary< BoundaryT, DomainTs >
- struct serif::discretization::domain::schema::validation::RelationValidationResult

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema
- namespace serif::discretization::domain::schema::validation

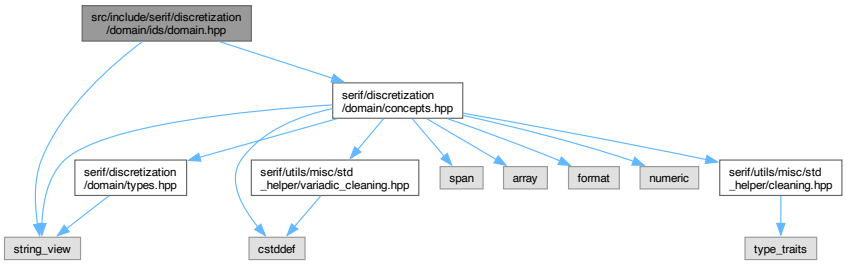
Enumerations

- enum class serif::discretization::domain::schema::validation::RelationValidationFailure

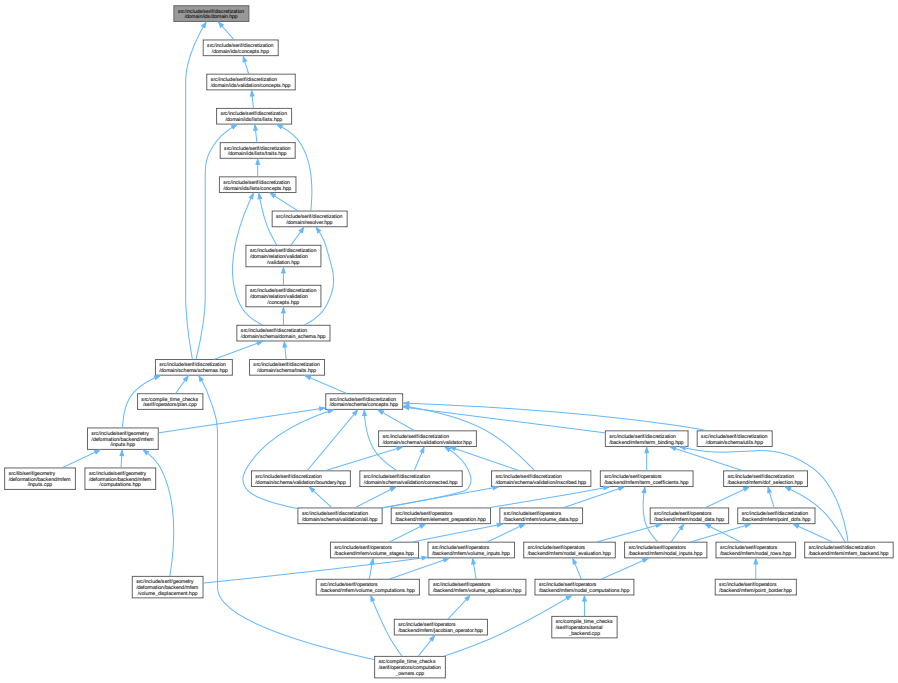
11.50 src/include/serif/discretization/domain/ids/domain.hpp File Reference

This file defines the DomainID type, which is used to bind a semantic volume (a Domain) to the integer element attribute MFEM stores in the mesh file.

#include <string_view>
#include "serif/discretization/domain/concepts.hpp"
Include dependency graph for domain.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >

  The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation.
- struct serif::discretization::domain::ids::DomainDescriptor

  The DomainDescriptor struct is a runtime representation of a domain, containing its name and integer attribute ID. This is used to project compile-time DomainID types into runtime values.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::ids

11.50.1 Detailed Description

This file defines the DomainID type, which is used to bind a semantic volume (a Domain) to the integer element attribute MFEM stores in the mesh file.

Author

Emily M. Boudreaux

For example, the following code defines a DomainID that associates the CoreDomain with the integer attribute 1 in the mesh file:

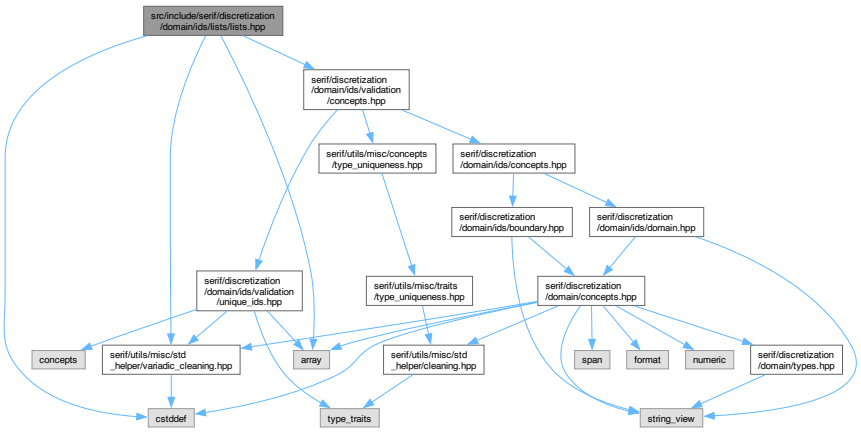
```
DomainID<CoreDomain, 1> coreDomainID;
```

Read this as ”elements with attribute 1 are the core”.

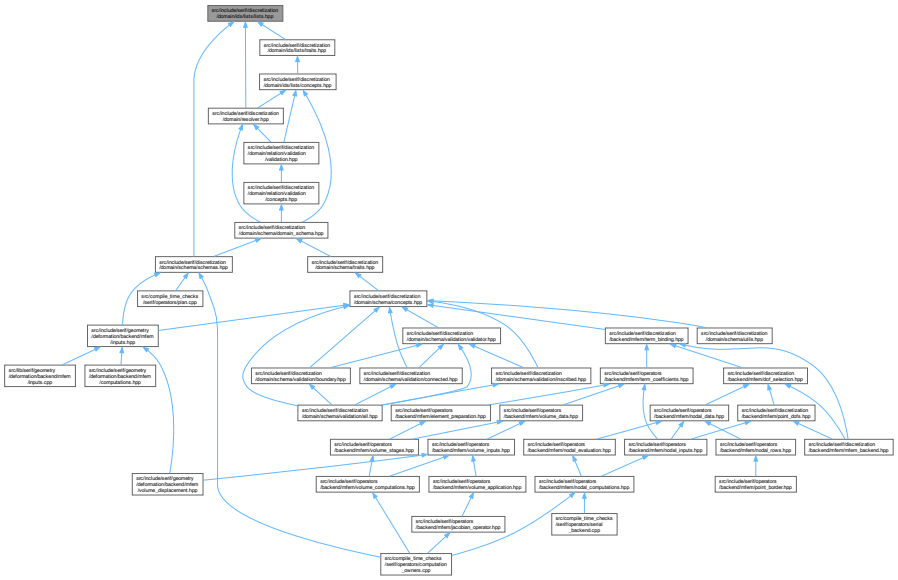
11.51 src/include/serif/discretization/domain/ids/lists/lists.hpp File Reference

This file defines the DomainIDList and BoundaryIDList types, specifically these types are used to project down compile time types to runtime values.

#include <array>
#include <cstddef>
#include ”serif/discretization/domain/ids/validation/concepts.hpp”
#include ”serif/utils/misc/std_helper/variadic_cleaning.hpp”
Include dependency graph for lists.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >
  A list of domain IDs in the discretization framework.
- struct serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs >
  A list of boundary IDs in the discretization framework.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::ids
- namespace serif::discretization::domain::ids::lists

Functions

- template<typename... Ts>
  consteval std::size_t serif::discretization::domain::ids::lists::CountVariadicArguments () noexcept

11.51.1 Detailed Description

This file defines the DomainIDList and BoundaryIDList types, specifically these types are used to project down compile time types to runtime values.

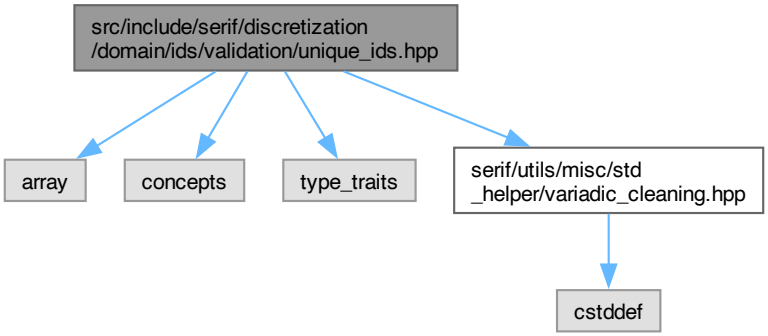
Author

Emily M. Boudreaux

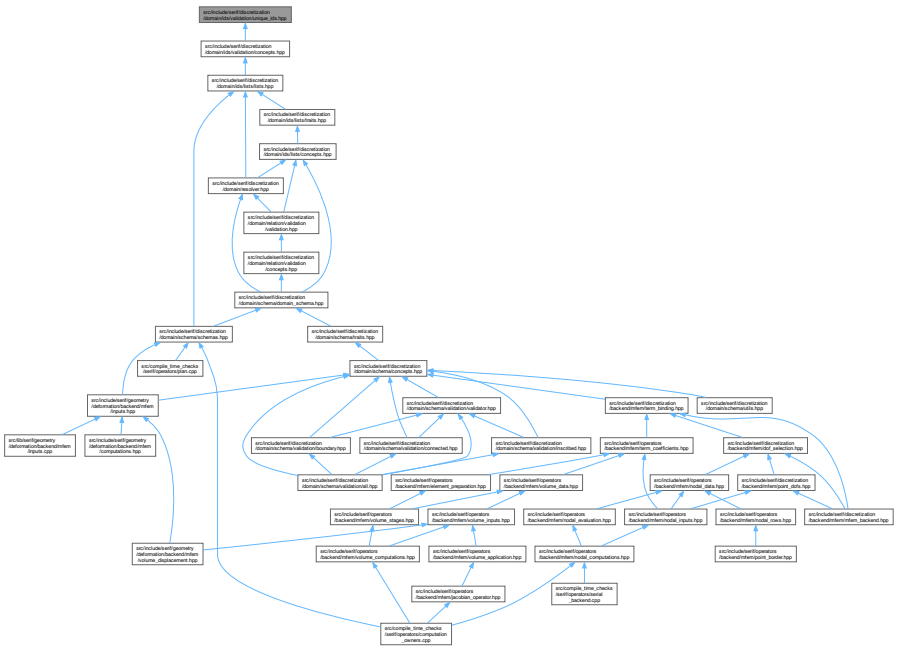
11.52 src/include/serif/discretization/domain/ids/validation/unique_↩ ids.hpp File Reference

This file defines a utility function to check if a pack of values are all unique. This is used to validate that domain and boundary IDs are distinct.

#include <array>
#include <concepts>
#include <type_traits>
#include "serif/utils/misc/std_helper/variadic_cleaning.hpp"
Include dependency graph for unique_ids.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::ids
- namespace serif::discretization::domain::ids::validation

Functions

- template<typename... Ts>
  requires (std::equality_comparable<Ts> && ...)
  consteval bool serif::discretization::domain::ids::validation::all_unique_values (Ts... values) noexcept

  Utility function to check if a pack of values are all unique.

11.52.1 Detailed Description

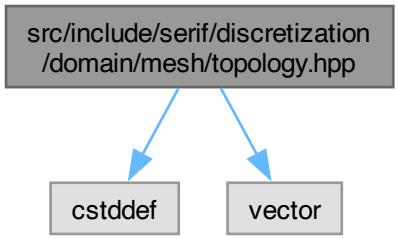
This file defines a utility function to check if a pack of values are all unique. This is used to validate that domain and boundary IDs are distinct.

Author

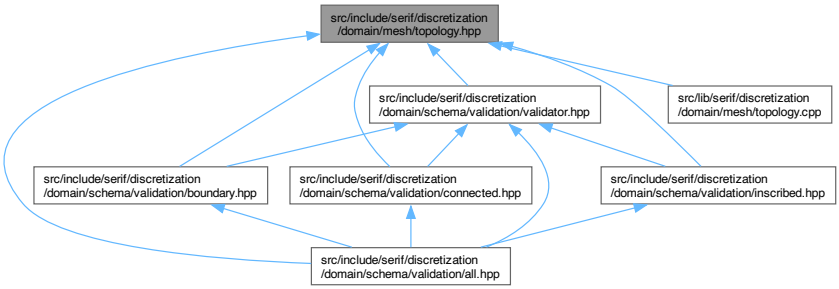
Emily M. Boudreaux

11.53 src/include/serif/discretization/domain/mesh/topology.hpp File Reference

#include <cstddef>
#include <vector>
Include dependency graph for topology.hpp:



This graph shows which files directly or indirectly include this file:



Classes

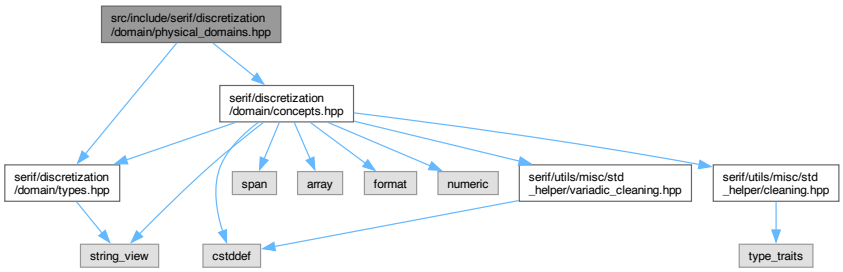
- struct serif::discretization::domain::mesh::FaceElements
- class serif::discretization::domain::mesh::MeshTopology

Namespaces

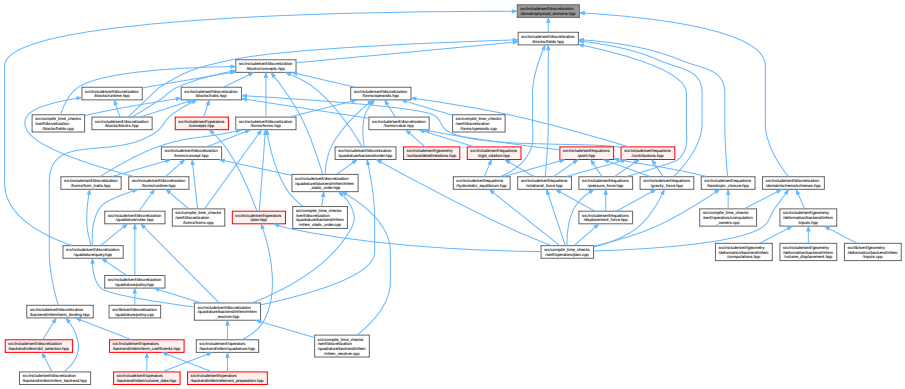
- namespace mfem
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::mesh

11.54 src/include/serif/discretization/domain/physical_domains.hpp File Reference

#include "serif/discretization/domain/types.hpp"
#include "serif/discretization/domain/concepts.hpp"
Include dependency graph for physical_domains.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

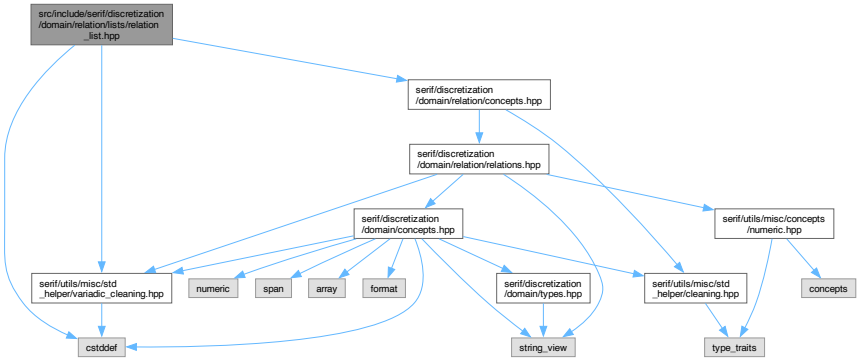
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain

Typedefs

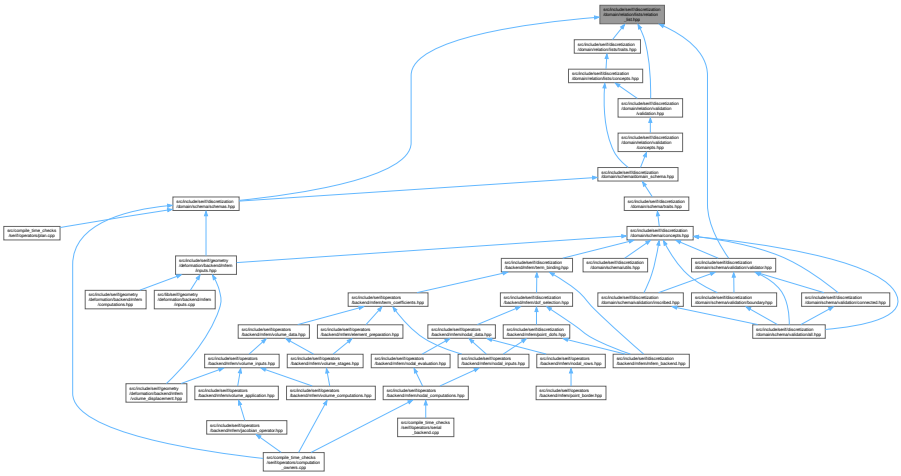
- using serif::discretization::domain::StellarDomains = DomainSet<CoreDomain, EnvelopeDomain>
- using serif::discretization::domain::AllDomains = DomainSet<CoreDomain, EnvelopeDomain, VacuumDomain>

11.55 src/include/serif/discretization/domain/relation/lists/relation_↩ list.hpp File Reference

#include <cstddef>
#include "serif/discretization/domain/relation/concepts.hpp"
#include "serif/utils/misc/std_helper/variadic_cleaning.hpp"
Include dependency graph for relation_list.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::relation::lists::RelationList< RelationTs >

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::relation
- namespace serif::discretization::domain::relation::lists

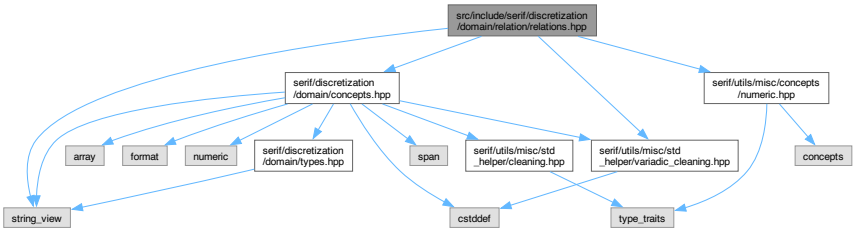
Functions

- template<typename... Ts>
  consteval std::size_t serif::discretization::domain::relation::lists::CountVariadicArguments () noexcept

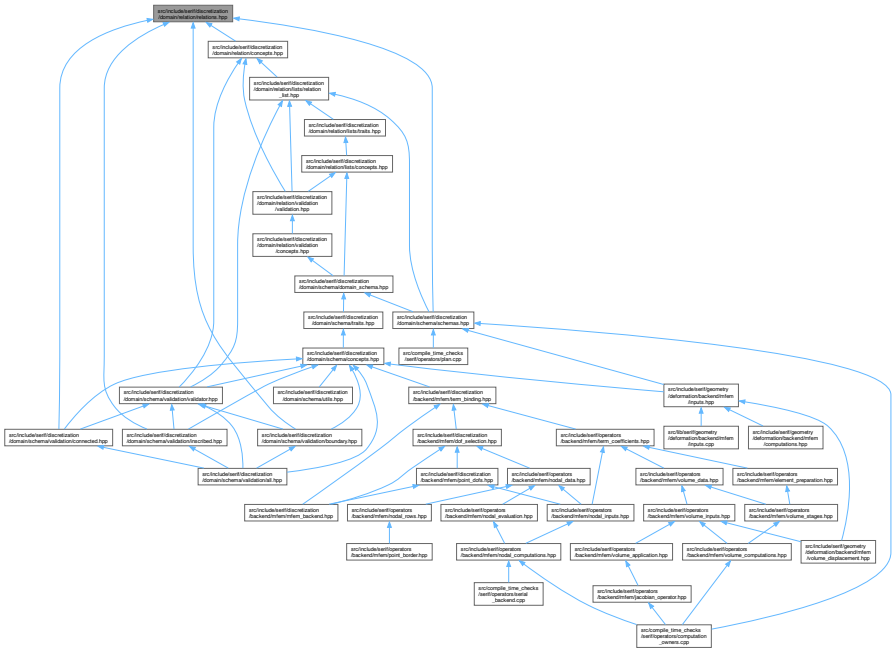
11.56 src/include/serif/discretization/domain/relation/relations.hpp File Reference

#include <string_view>
#include "serif/discretization/domain/concepts.hpp"
#include "serif/utils/misc/concepts/numeric.hpp"

#include ”serif/utils/misc/std_helper/variadic_cleaning.hpp”
Include dependency graph for relations.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::relation::DomainRelation
- struct serif::discretization::domain::relation::Inscribed< A, B >
- struct serif::discretization::domain::relation::FullyConnected< A >
- struct serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >

Namespaces

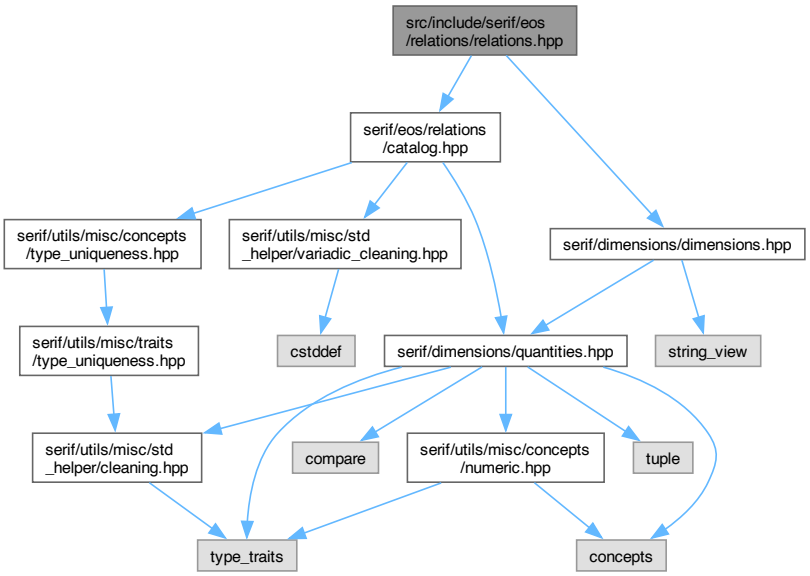
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::relation

Functions

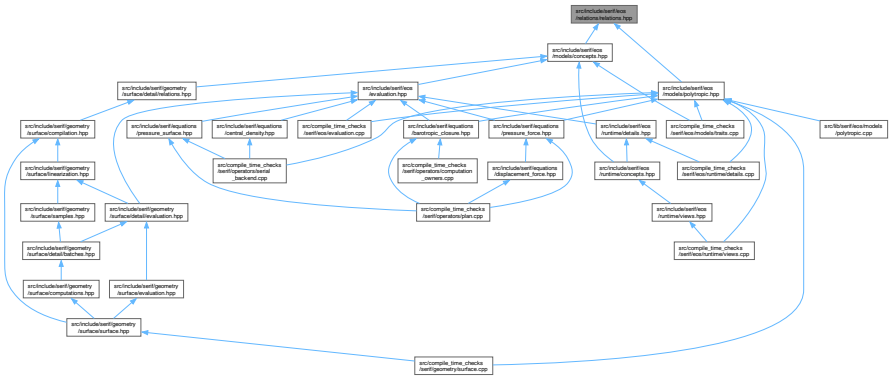
- template<typename... Ts>
  consteval std::size_t serif::discretization::domain::relation::CountVariadicArguments () noexcept

11.57 src/include/serif/eos/relations/relations.hpp File Reference

#include "serif/eos/relations/catalog.hpp"
#include "serif/dimensions/dimensions.hpp"
Include dependency graph for relations.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

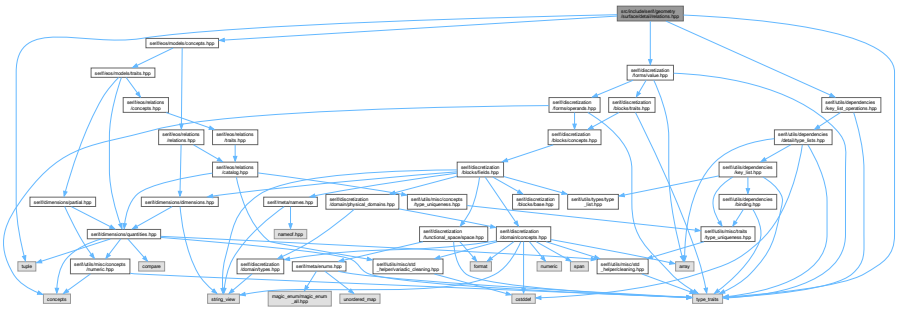
- namespace serif
- namespace serif::eos
- namespace serif::eos::relations

Typedefs

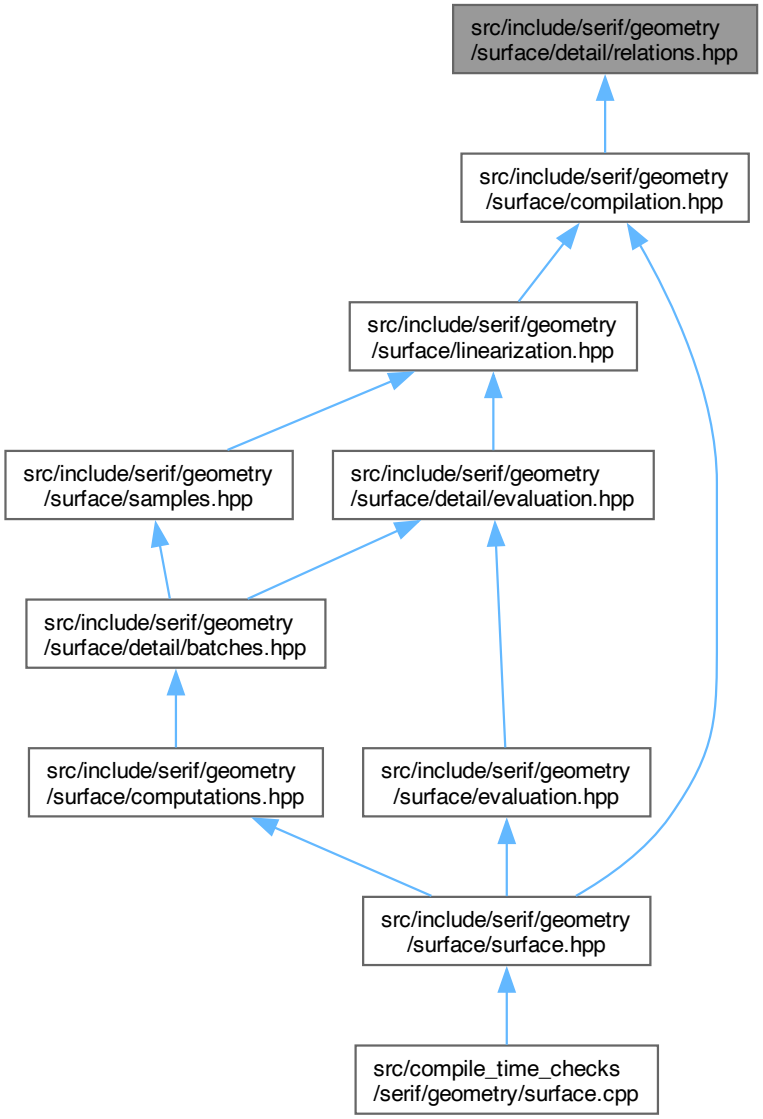
- using serif::eos::relations::PressureFromDensity
- using serif::eos::relations::PressureFromSpecificEnthalpy
- using serif::eos::relations::SpecificEnthalpyFromDensity
- using serif::eos::relations::SpecificEnthalpyFromPressure
- using serif::eos::relations::DensityFromSpecificEnthalpy

11.58 src/include/serif/geometry/surface/detail/relations.hpp File Reference

#include <tuple>
#include <type_traits>
#include "serif/discretization/forms/value.hpp"
#include "serif/eos/models/concepts.hpp"
#include "serif/utils/dependencies/key_list_operations.hpp"
Include dependency graph for relations.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::surface::detail::AreSurfaceBindings< Bindings >
- struct serif::geometry::surface::detail::AreSurfaceBindings< utils::dependencies::Bindings< Entries... > >
- struct serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > >
- struct serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCata
- struct serif::geometry::surface::detail::SurfaceOperands< utils::dependencies::KeyList< Terms... > >
- struct serif::geometry::surface::detail::UniqueTerms< utils::types::TypeList< Terms... > >

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::surface
- namespace serif::geometry::surface::detail

Concepts

- concept serif::geometry::surface::SurfaceScalarTerm

  Scalar thermodynamic data may come from any declared scalar space; boundary enforcement is a separate backend capability.
- concept serif::geometry::surface::SurfaceStateBindings

  Reuse dependency bindings, retaining distinct quantity roles even when they refer to one term.
- concept serif::geometry::surface::SurfaceResidualRow

  The quantity enforcing the boundary must have an actual equation row, not just sampled values.

Typedefs

- template<typename Bindings, typename Quantities>
  using serif::geometry::surface::detail::BoundTerms = typename UniqueTerms<utils::dependencies::ResolvedTargets<Bin
  Quantities>>::Type

Functions

- template<typename EOS, typename Relation, typename... Quantities>
  consteval bool serif::geometry::surface::detail::supports_partials (utils::dependencies::KeyList< Quantities... >)

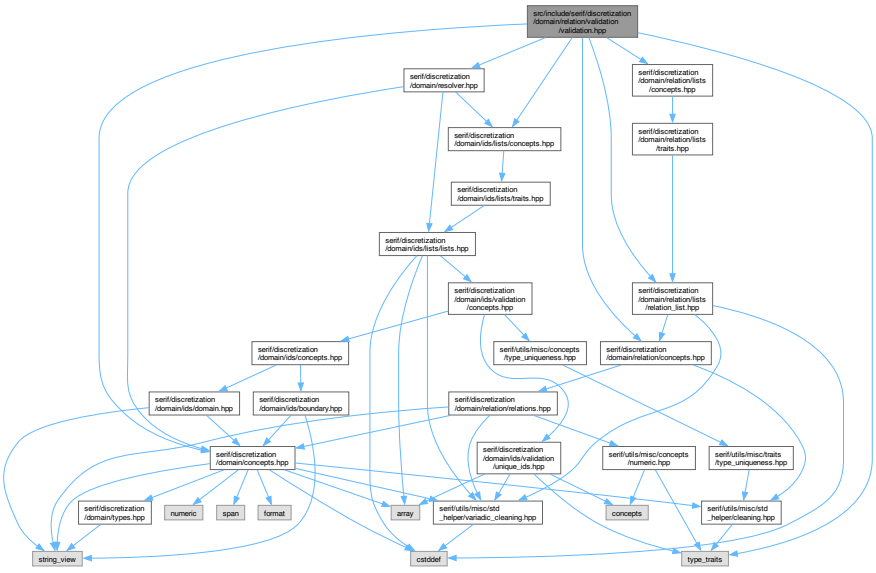
  Check only the derivatives of unknown state; prescribed pressure has no Newton column.
- template<typename EOS, typename Bindings, typename Carrier, typename Relation>
  consteval bool serif::geometry::surface::detail::matches_pressure_relation ()

  A resolved carrier must be supplied by one pressure-input relation, with all remaining inputs and partials available.

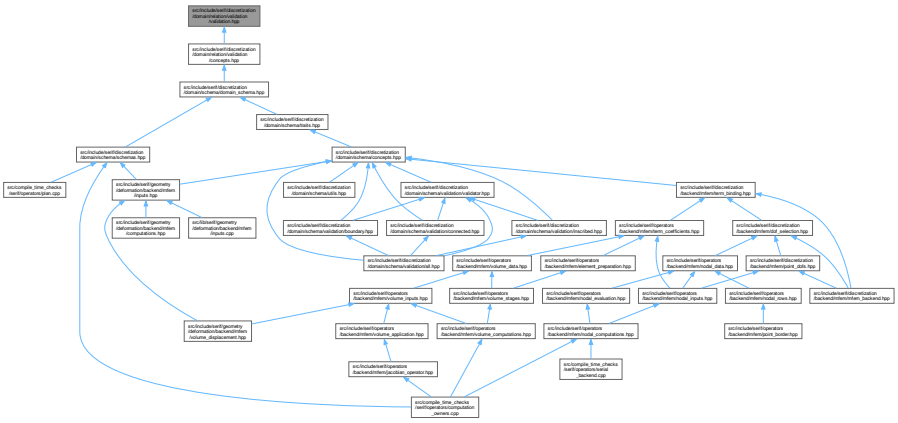
11.59 src/include/serif/discretization/domain/relation/validation/validation.hpp File Reference

#include <type_traits>
#include "serif/discretization/domain/concepts.hpp"
#include "serif/discretization/domain/ids/lists/concepts.hpp"
#include "serif/discretization/domain/relation/concepts.hpp"
#include "serif/discretization/domain/relation/lists/concepts.hpp"
#include "serif/discretization/domain/relation/lists/relation_list.hpp"

#include "serif/discretization/domain/resolver.hpp"
Include dependency graph for validation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

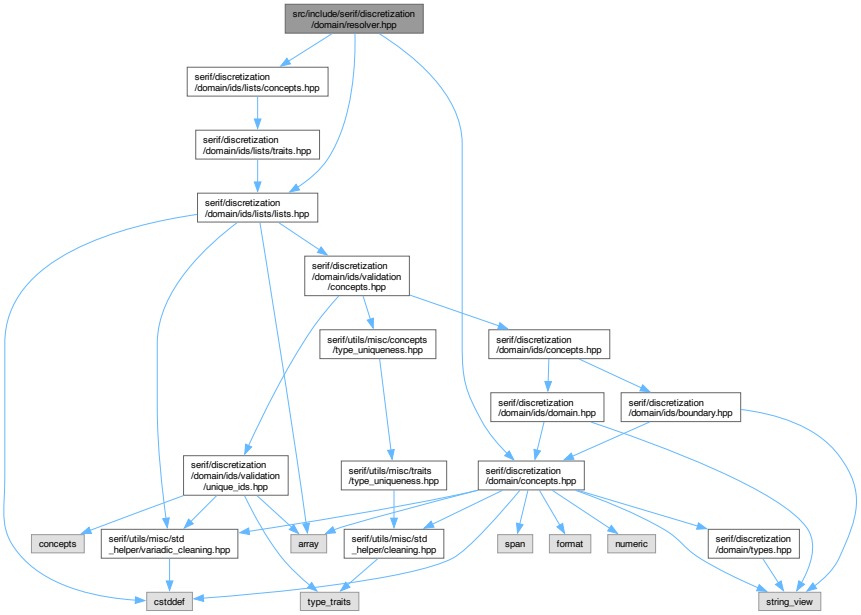
- struct serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< FullyConnected< DomainT >
- struct serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< Inscribed< InnerDomainT, O
- struct serif::discretization::domain::relation::validation::RelationUsesRegisteredEntities< DomainBoundary< Boundary
- struct serif::discretization::domain::relation::validation::RelationsUseRegisteredEntities< DomainIDsT, BoundaryIDsT,

Namespaces

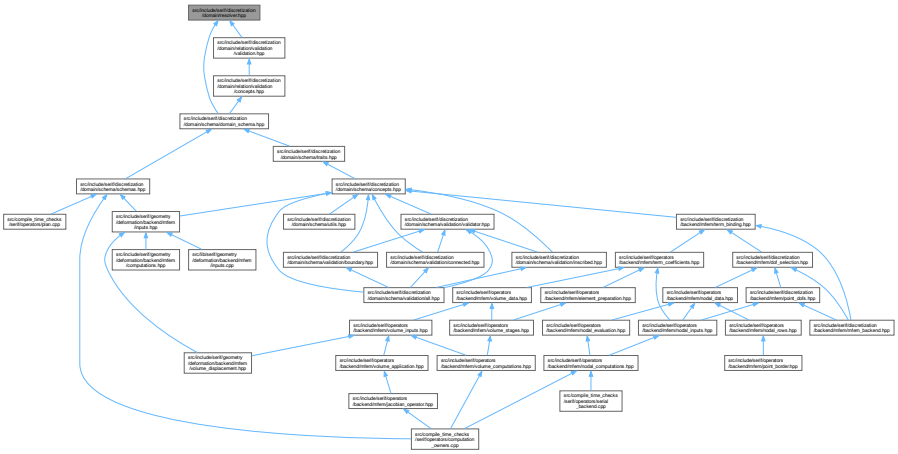
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::relation
- namespace serif::discretization::domain::relation::validation

11.60 src/include/serif/discretization/domain/resolver.hpp File Reference

#include ”serif/discretization/domain/concepts.hpp”
#include ”serif/discretization/domain/ids/lists/concepts.hpp”
#include ”serif/discretization/domain/ids/lists/lists.hpp”
Include dependency graph for resolver.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > >
- struct serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< Doma
- struct serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... >
- struct serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList<

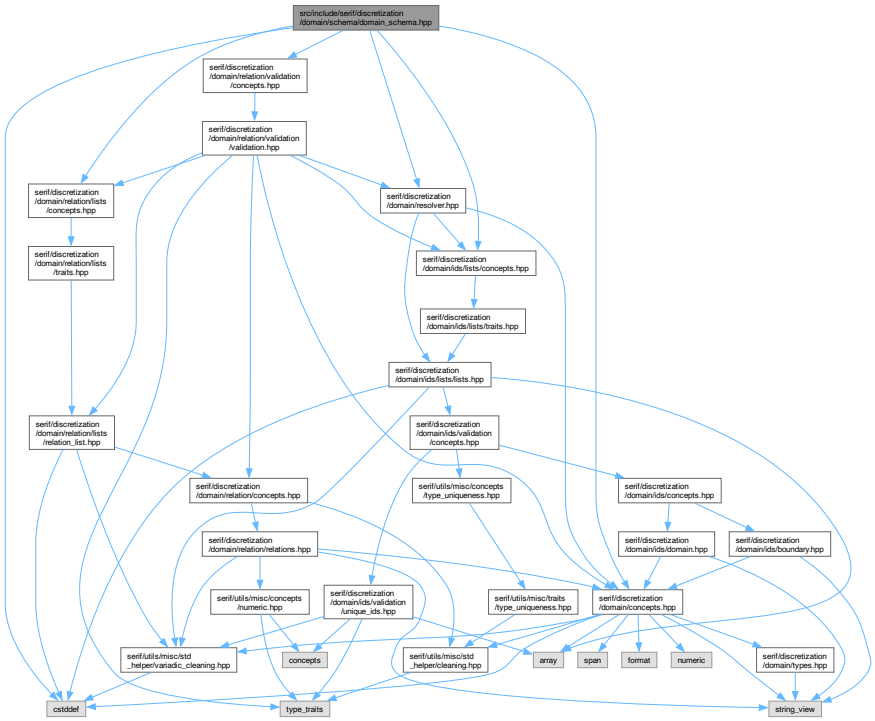
Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain

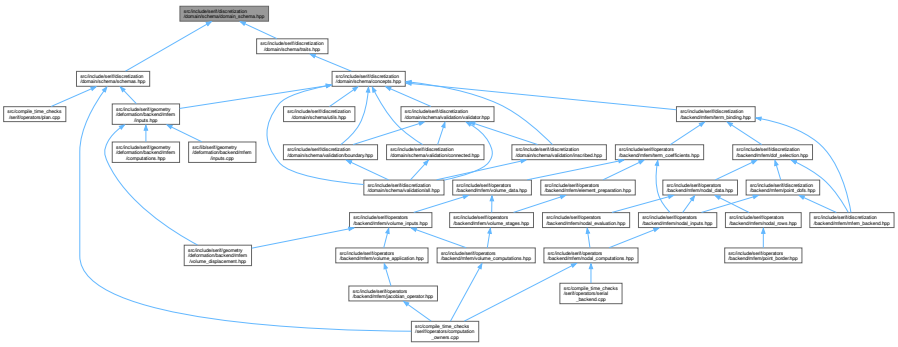
11.61 src/include/serif/discretization/domain/schema/domain_↩ schema.hpp File Reference

#include <cstddef>
#include "serif/discretization/domain/concepts.hpp"
#include "serif/discretization/domain/ids/lists/concepts.hpp"
#include "serif/discretization/domain/relation/lists/concepts.hpp"
#include "serif/discretization/domain/relation/validation/concepts.hpp"

#include ”serif/discretization/domain/resolver.hpp”
Include dependency graph for domain_schema.hpp:



This graph shows which files directly or indirectly include this file:



Classes

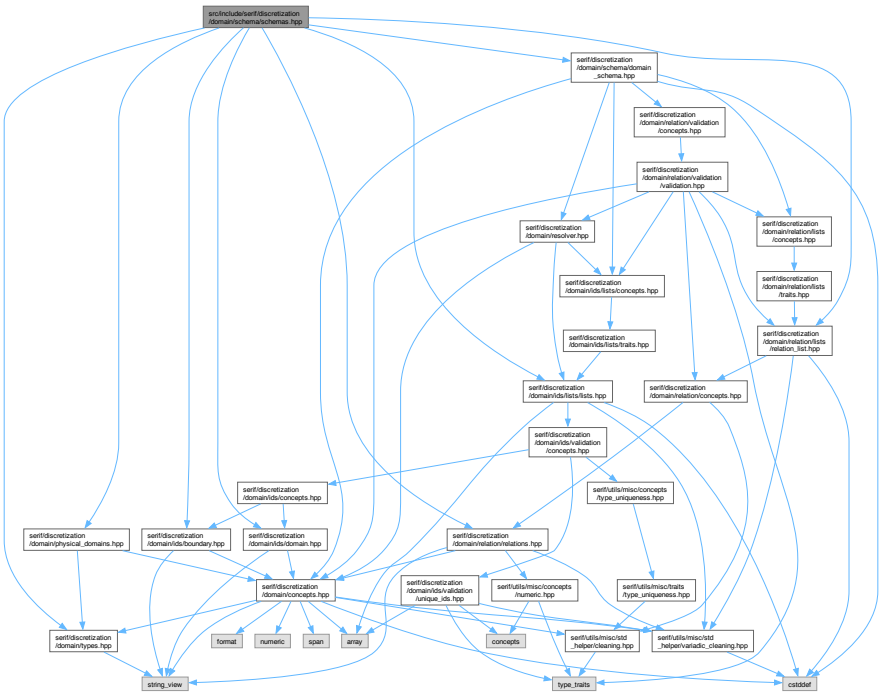
- struct serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >

Namespaces

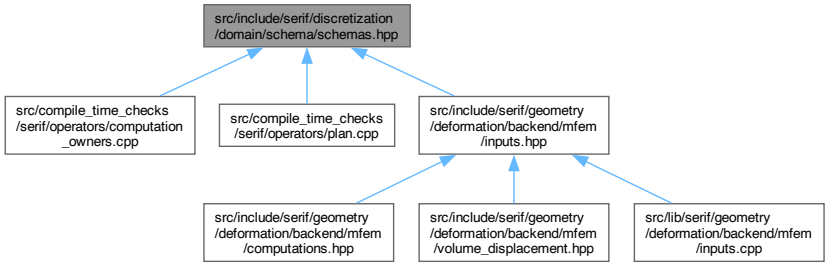
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema

11.62 src/include/serif/discretization/domain/schema/schemas.hpp File Reference

#include ”serif/discretization/domain/ids/boundary.hpp”
#include ”serif/discretization/domain/ids/domain.hpp”
#include ”serif/discretization/domain/ids/lists/lists.hpp”
#include ”serif/discretization/domain/physical_domains.hpp”
#include ”serif/discretization/domain/relation/lists/relation_list.hpp”
#include ”serif/discretization/domain/relation/relations.hpp”
#include ”serif/discretization/domain/schema/domain_schema.hpp”
#include ”serif/discretization/domain/types.hpp”
Include dependency graph for schemas.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::schema::DomainID< DomainT, IdentifierValue >

  The DomainID struct is a compile-time type that associates a semantic domain (a Domain) with an integer attribute in the mesh file. This allows the discretization framework to identify which elements in the mesh correspond to which semantic domains in the simulation.
- struct serif::discretization::domain::schema::BoundaryID< BoundaryT, IdentifierValue >

  The BoundaryID struct is a compile-time type that associates a semantic boundary (a Boundary) with an integer attribute in the mesh file. This allows the discretization framework to identify which boundary elements in the mesh correspond to which semantic boundaries in the simulation.
- struct serif::discretization::domain::schema::DomainIDList< DomainIDTs >

  A list of domain IDs in the discretization framework.
- struct serif::discretization::domain::schema::BoundaryIDList< BoundaryIDTs >

  A list of boundary IDs in the discretization framework.
- struct serif::discretization::domain::schema::RelationList< RelationTs >

Namespaces

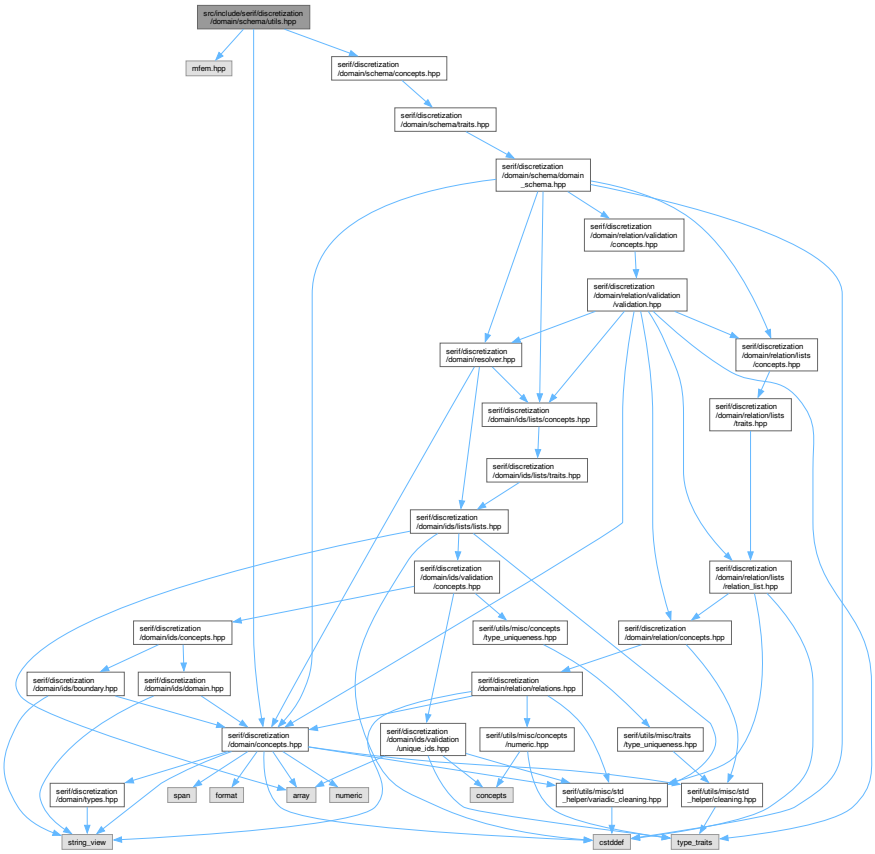
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema

Typedefs

- using serif::discretization::domain::schema::CoreEnvelopeVacuumDomainSchema

11.63 src/include/serif/discretization/domain/schema/utils.hpp File Reference

#include <mfem.hpp>
#include "serif/discretization/domain/concepts.hpp"
#include "serif/discretization/domain/schema/concepts.hpp"
Include dependency graph for utils.hpp:



Namespaces

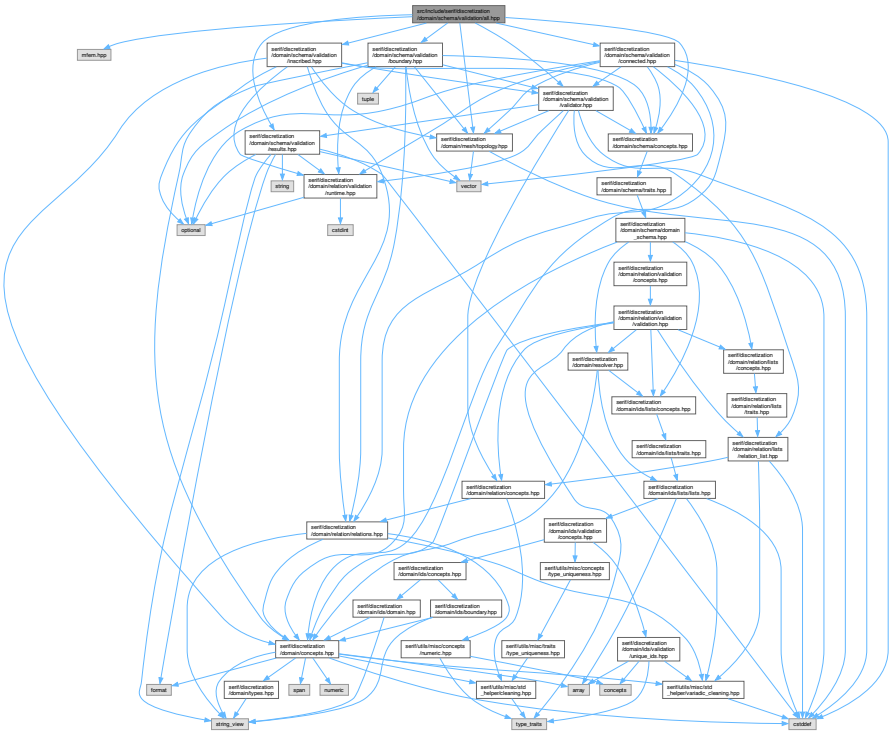
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema

Functions

- template<IsDomainOrSet DomainT, IsSchema SchemaT>
  mfem::Array< int > serif::discretization::domain::schema::make_attribute_marker (const mfem::Mesh &mesh)

11.64 src/include/serif/discretization/domain/schema/validation/all.hpp File Reference

#include <mfem.hpp>
#include "serif/discretization/domain/mesh/topology.hpp"
#include "serif/discretization/domain/schema/concepts.hpp"
#include "serif/discretization/domain/schema/validation/results.hpp"
#include "serif/discretization/domain/schema/validation/validator.hpp"
#include "serif/discretization/domain/schema/validation/boundary.hpp"
#include "serif/discretization/domain/schema/validation/connected.hpp"
#include "serif/discretization/domain/schema/validation/inscribed.hpp"
Include dependency graph for all.hpp:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema
- namespace serif::discretization::domain::schema::validation

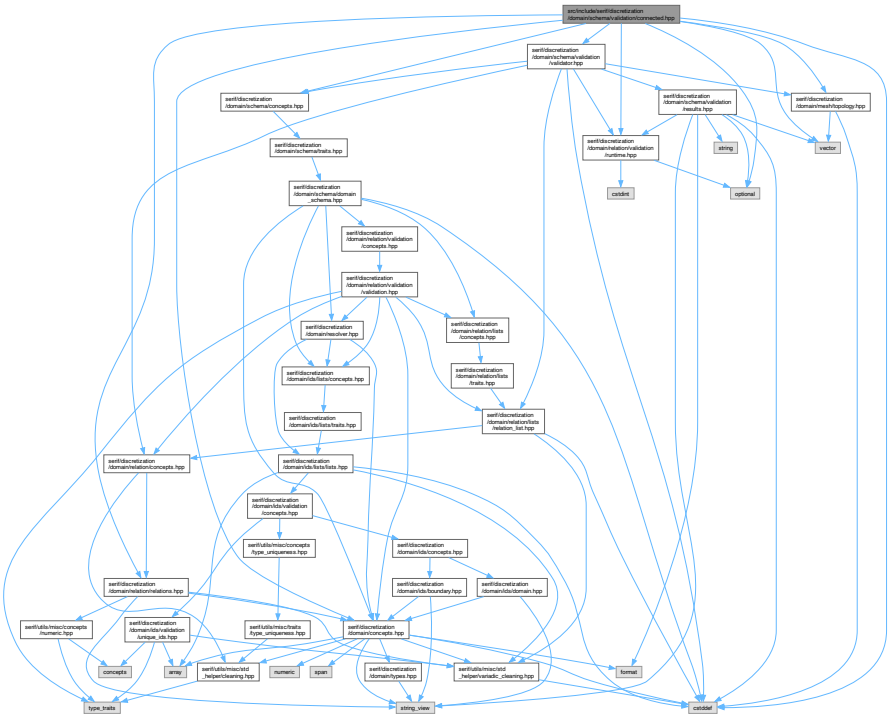
Functions

- template<IsSchema SchemaT> SchemaValidationResult serif::discretization::domain::schema::validation::validate_schema (const mesh::MeshTopology &topology)

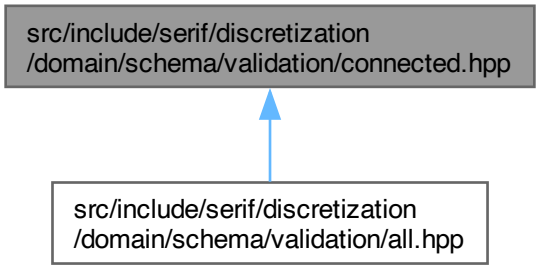
- template<IsSchema SchemaT>
  SchemaValidationResult serif::discretization::domain::schema::validation::validate_schema (const mfem::Mesh &mfemMesh)

11.65 src/include/serif/discretization/domain/schema/validation/connected.hpp File Reference

#include <cstddef>
#include <optional>
#include <vector>
#include "serif/discretization/domain/concepts.hpp"
#include "serif/discretization/domain/mesh/topology.hpp"
#include "serif/discretization/domain/relation/relations.hpp"
#include "serif/discretization/domain/relation/validation/runtime.hpp"
#include "serif/discretization/domain/schema/concepts.hpp"
#include "serif/discretization/domain/schema/validation/validator.hpp"
Include dependency graph for connected.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::schema::validation::RelationValidator< FullyConnected< DomainT > >
- struct serif::discretization::domain::schema::validation::FullyConnected< A >

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema
- namespace serif::discretization::domain::schema::validation

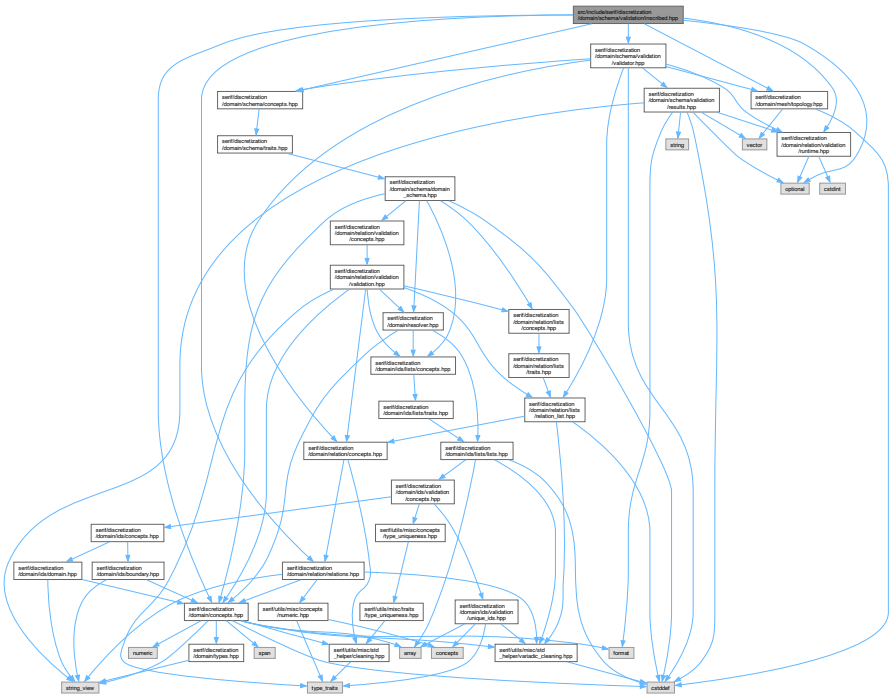
Enumerations

- enum class serif::discretization::domain::schema::validation::RelationValidationFailure

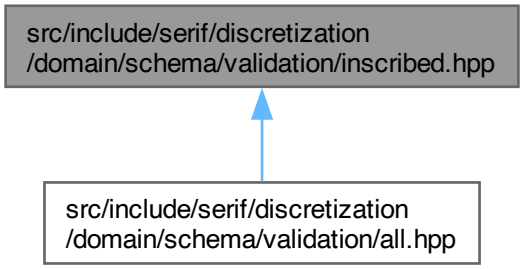
11.66 src/include/serif/discretization/domain/schema/validation/inscribed.hpp File Reference

#include <optional>
#include "serif/discretization/domain/concepts.hpp"
#include "serif/discretization/domain/mesh/topology.hpp"
#include "serif/discretization/domain/relation/relations.hpp"
#include "serif/discretization/domain/relation/validation/runtime.hpp"
#include "serif/discretization/domain/schema/concepts.hpp"

#include "serif/discretization/domain/schema/validation/validator.hpp"
Include dependency graph for inscribed.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::schema::validation::RelationValidator< Inscribed< InnerT, OuterT > >
- struct serif::discretization::domain::schema::validation::Inscribed< A, B >

Namespaces

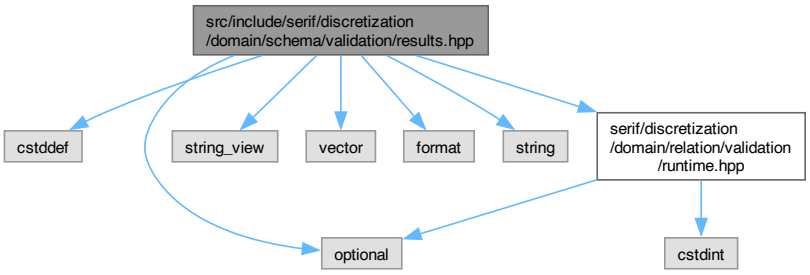
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema
- namespace serif::discretization::domain::schema::validation

Enumerations

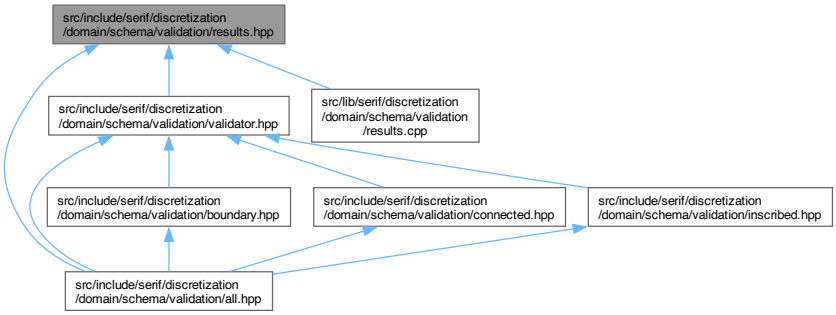
- enum class serif::discretization::domain::schema::validation::RelationValidationFailure

11.67 src/include/serif/discretization/domain/schema/validation/results.hpp File Reference

#include <cstddef>
#include <optional>
#include <string_view>
#include <vector>
#include <format>
#include <string>
#include "serif/discretization/domain/relation/validation/runtime.hpp"
Include dependency graph for results.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::schema::validation::SchemaRelationValidationResult
- struct serif::discretization::domain::schema::validation::SchemaValidationResult
- struct std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult >

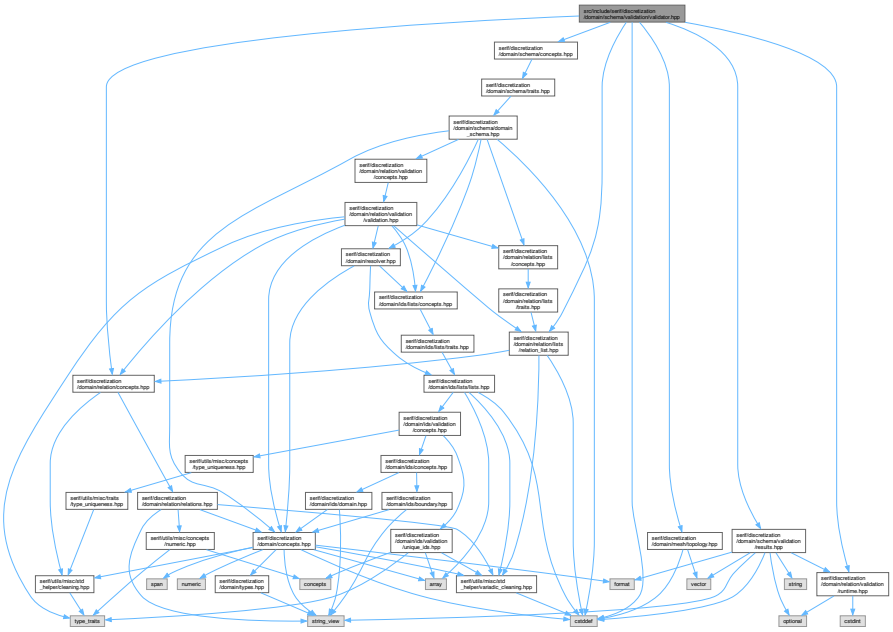
Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema
- namespace serif::discretization::domain::schema::validation

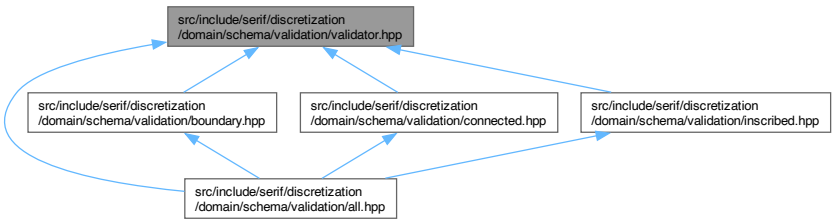
11.68 src/include/serif/discretization/domain/schema/validation/validator.hpp File Reference

#include <cstddef>
#include "serif/discretization/domain/mesh/topology.hpp"
#include "serif/discretization/domain/relation/concepts.hpp"
#include "serif/discretization/domain/relation/lists/relation_list.hpp"
#include "serif/discretization/domain/relation/validation/runtime.hpp"
#include "serif/discretization/domain/schema/concepts.hpp"

#include ”serif/discretization/domain/schema/validation/results.hpp”
Include dependency graph for validator.hpp:



This graph shows which files directly or indirectly include this file:



Classes

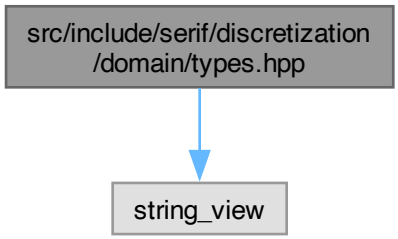
- struct serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, relation::lists::RelationLis

Namespaces

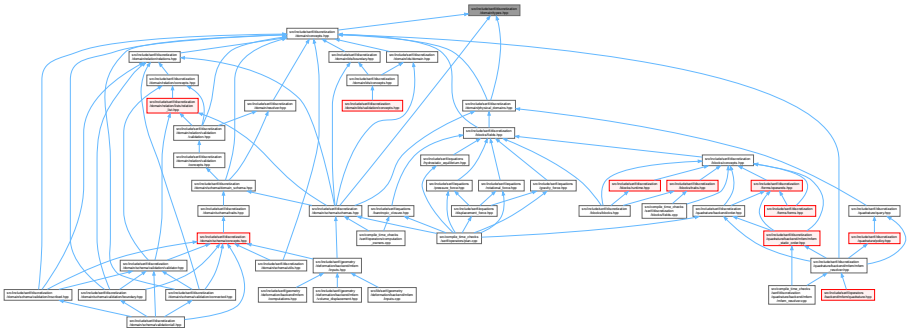
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema
- namespace serif::discretization::domain::schema::validation

11.69 src/include/serif/discretization/domain/types.hpp File Reference

#include <string_view>
Include dependency graph for types.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::Domain
- struct serif::discretization::domain::CoreDomain
- struct serif::discretization::domain::EnvelopeDomain
- struct serif::discretization::domain::VacuumDomain
- struct serif::discretization::domain::Boundary
- struct serif::discretization::domain::StellarSurfaceBoundary
- struct serif::discretization::domain::InfinitySurfaceBoundary

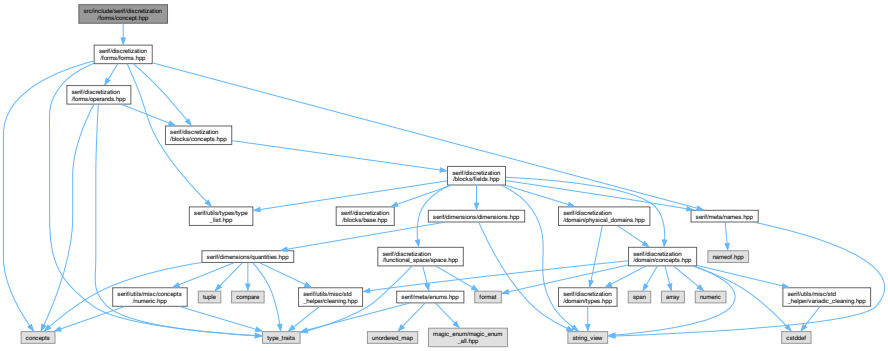
Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain

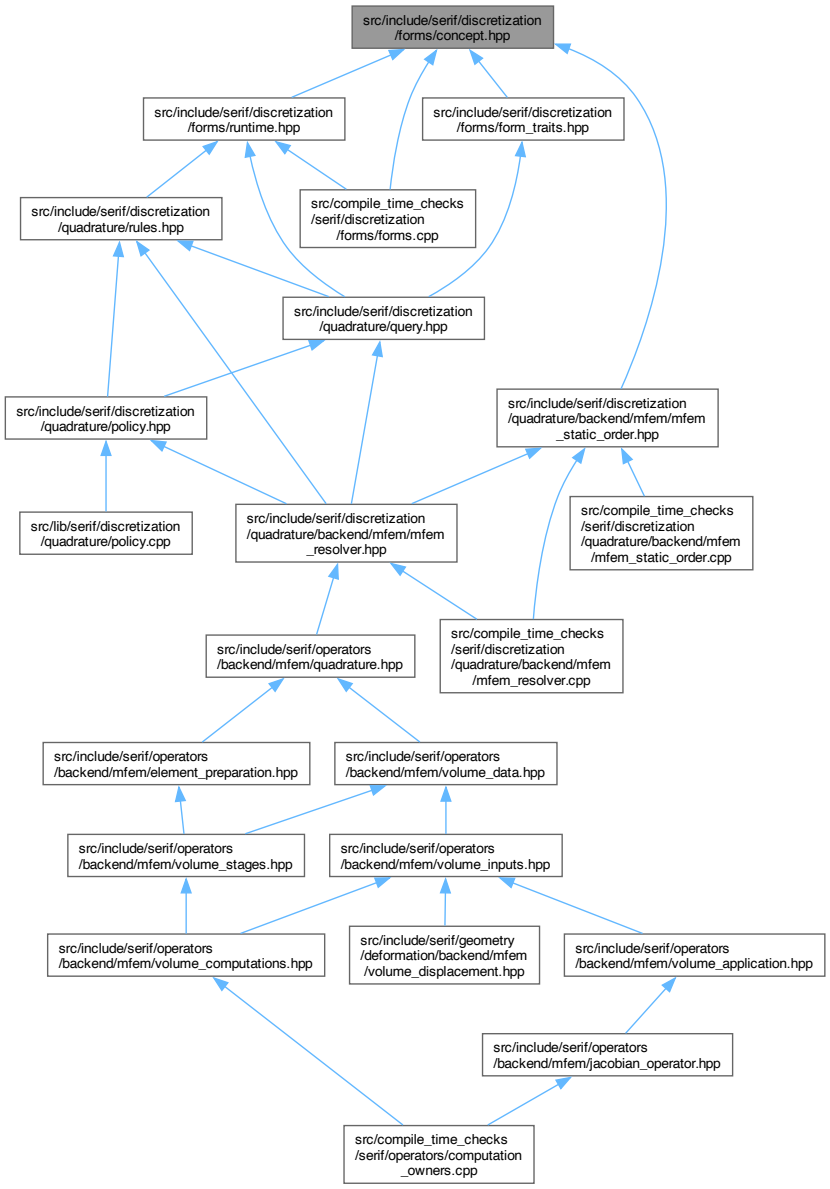
11.70 src/include/serif/discretization/forms/concept.hpp File Reference

This file defines concepts for forms in the discretization framework.

#include "serif/discretization/forms/forms.hpp"
Include dependency graph for concept.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::forms

Concepts

- concept serif::discretization::forms::Form

Concept for types that are forms in the discretization framework. In order satisfy this concept a type must derive from the form base class and must define a member type called operand_types which is intended to be a type list of the trial and test terms for the form.

11.70.1 Detailed Description

This file defines concepts for forms in the discretization framework.

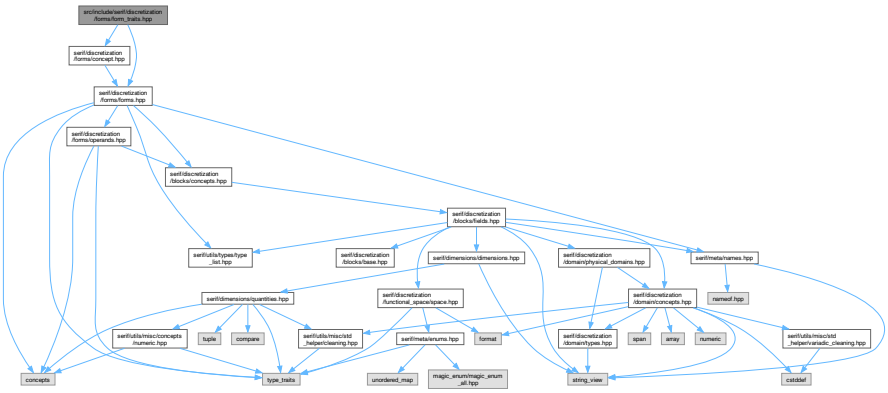
Author

Emily M. Boudreaux

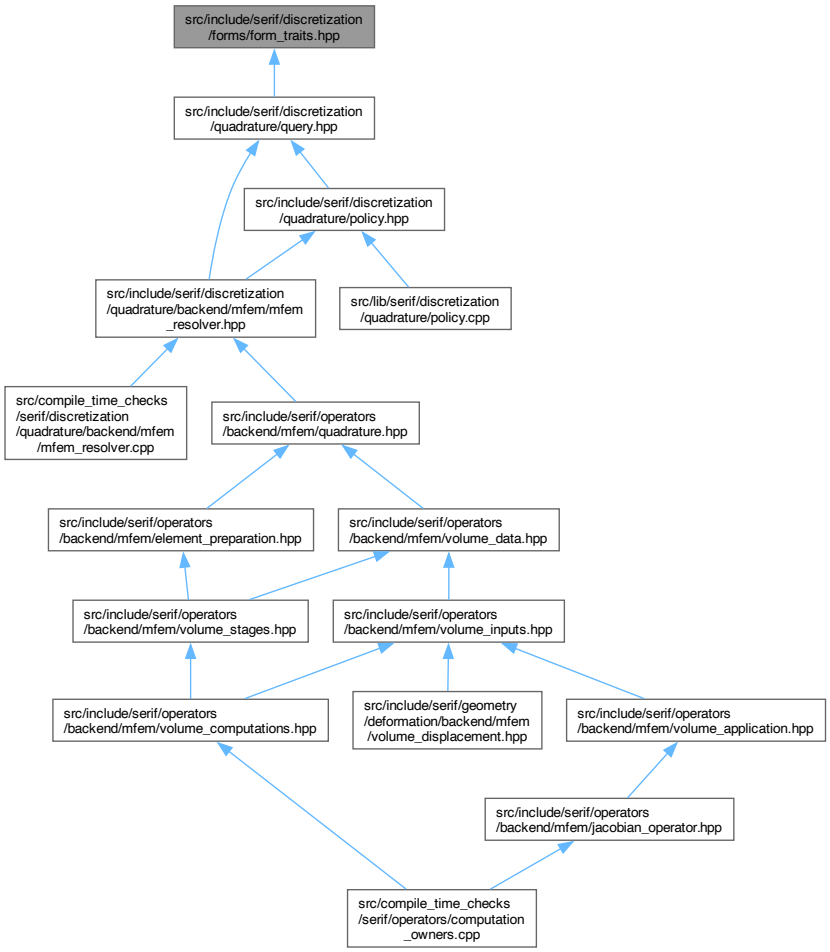
11.71 src/include/serif/discretization/forms/form_traits.hpp File Reference

Compile-time traits for forms, used to extract types and properties from template parameters.

#include "serif/discretization/forms/concept.hpp"
#include "serif/discretization/forms/forms.hpp"
Include dependency graph for form_traits.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::forms::FormTraits< FormT >

  Compile-time traits for forms, used to extract types and properties from template parameters. This struct is specialized for each form type to provide specific traits such as trial and test order reductions.
- struct serif::discretization::forms::FormTraits< mass< TrialT, TestT > >

  Specialization of FormTraits for the mass form, which has no order reduction for either trial or test terms.
- struct serif::discretization::forms::FormTraits< divergence< TrialT, TestT > >

  Specialization of FormTraits for the divergence form, which has an order reduction of 1, to satisfy the Inf-Sup condition, for the trial term and no order reduction for the test term.
- struct serif::discretization::forms::FormTraits< source< TrialT, TestT > >

  Specialization of FormTraits for the source form, which has no order reduction for either trial or test terms.
- struct serif::discretization::forms::FormTraits< force< TrialT, TestT > >

  Specialization of FormTraits for the force form, which has no order reduction for either trial or test terms.
- struct serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > >

Specialization of FormTraits for the boundary_flux form, which has no order reduction for either trial or test terms.

- struct serif::discretization::forms::FormTraits< projection< TrialT, TestT > >

  Specialization of FormTraits for the projection form, which has no order reduction for either trial or test terms.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::forms

11.71.1 Detailed Description

Compile-time traits for forms, used to extract types and properties from template parameters.

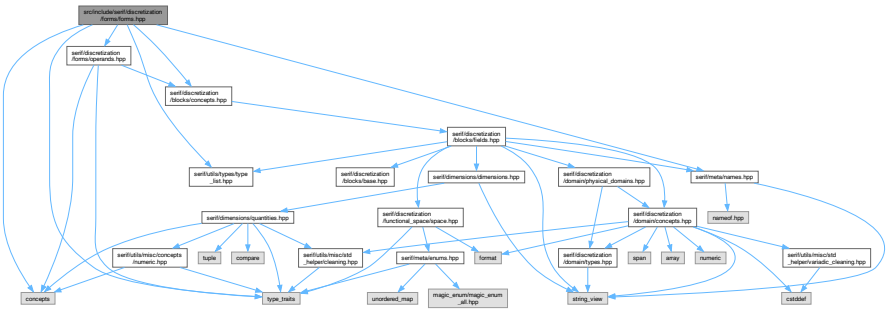
Author

Emily M. Boudreaux

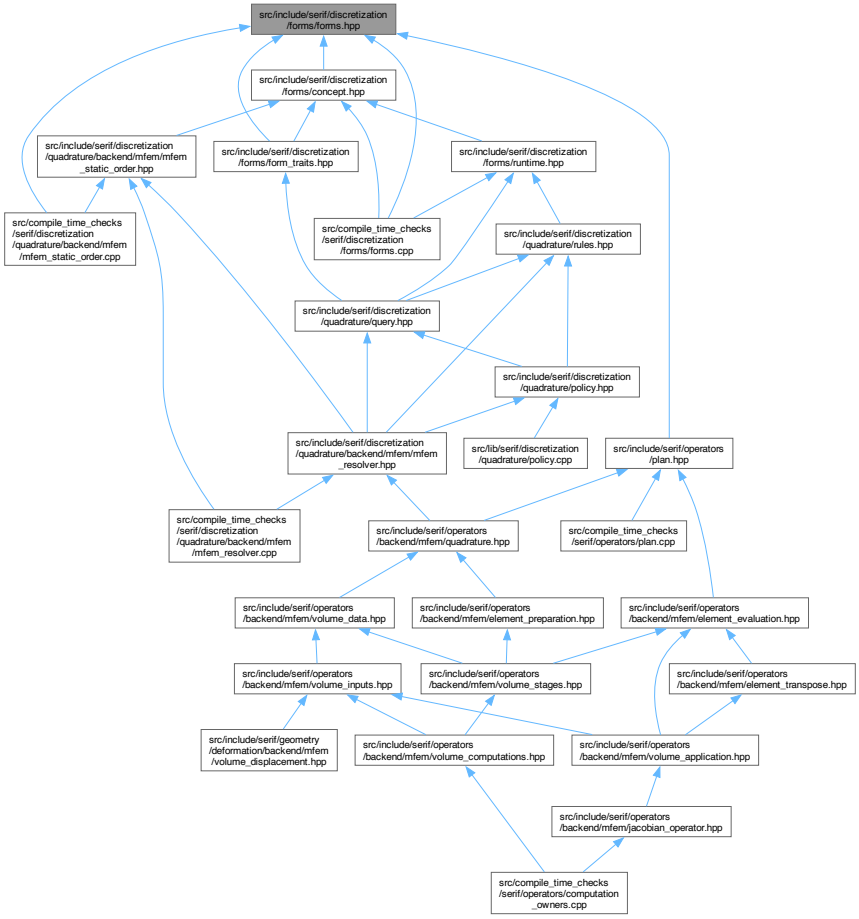
11.72 src/include/serif/discretization/forms/forms.hpp File Reference

This file defines the forms used in the discretization framework, including mass, divergence, source, force, boundary flux, and projection forms.

#include <concepts>
#include <type_traits>
#include "serif/discretization/blocks/concepts.hpp"
#include "serif/discretization/forms/operands.hpp"
#include "serif/utils/types/type_list.hpp"
#include "serif/meta/names.hpp"
Include dependency graph for forms.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::forms::form

  The form struct is a tag type that is used to identify a form in the discretization. It is used to group together forms that belong to the same mathematical operation. Each form should have a unique name and a list of operand types that belong to it.
- struct serif::discretization::forms::mass< TrialT, TestT >

  The mass form represents the mass matrix operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the mass operation.
- struct serif::discretization::forms::divergence< TrialT, TestT >

  The divergence form represents the divergence operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the divergence operation.
- struct serif::discretization::forms::source< TrialT, TestT >

  The source form represents the source term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the source operation.
- struct serif::discretization::forms::force< TrialT, TestT >

The force form represents the force term in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the force operation.

- struct serif::discretization::forms::boundary_flux< TrialT, TestT >

  The boundary_flux form represents the boundary flux operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the boundary flux operation.

- struct serif::discretization::forms::projection< TrialT, TestT >

  The projection form represents the projection operation in the discretization framework. It takes a trial term and a test term as template parameters and defines the operand types for the projection operation.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::forms

11.72.1 Detailed Description

This file defines the forms used in the discretization framework, including mass, divergence, source, force, boundary flux, and projection forms.

Author

Emily M. Boudreaux

Note

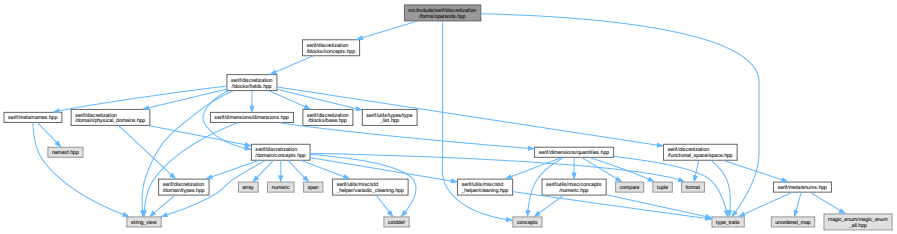
If you are a developer tasked with the likely horrid task of adding some new equation that cannot be represented within the current framework that SERiF provides, this should probably be your first stop. This file essentially defines the mathematical operations that SERiF can perform. As you can see below none of these forms are particularly complex, and they certainly do not actually describe how to perform these operations. Rather, these form the base of a chain of compile-time probability that SERiF knows how to do something. Effectively this file is a contract with the rest of the code that says "If someone asks for the divergence of a trial field against some test field that is okay because I have claimed in this file that I know how to do that". The actual implementation lives elsewhere.

Todo Come back and fill in more information about elsewhere. If you are the developer from above and you are reading this todo, then I am, and I mean this truly, very sorry....

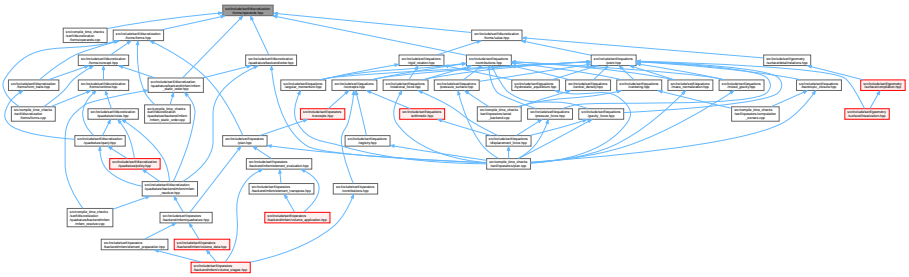
11.73 src/include/serif/discretization/forms/operands.hpp File Reference

This file defines the Operand type, which represents a term and an operation to be applied to that term in the context of a form.

#include <concepts>
#include <type_traits>
#include "serif/discretization/blocks/concepts.hpp"
Include dependency graph for operands.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::forms::Operation

  The Operation struct defines a set of operations that can be applied to terms in the context of a form. These operations include value, gradient, divergence, curl, and normal trace. Similar to the statement I made at the top of forms.hpp, if you are a developer looking to extend the mathematical capabilities of SERiF this is a place you should be looking early on.
- struct serif::discretization::forms::Operation::Value

  Operation tag for evaluating a value.
- struct serif::discretization::forms::Operation::Gradient

  Operation tag for evaluating a gradient.
- struct serif::discretization::forms::Operation::Divergence

  Operation tag for evaluating a divergence.
- struct serif::discretization::forms::Operation::Curl

  Operation tag for evaluating a curl.
- struct serif::discretization::forms::Operation::NormalTrace

  Operation tag for evaluating the normal flux along a boundary.
- struct serif::discretization::forms::Operand< TermT, OperationT >

  Compile time representation of some operand. An operand associates a term to an operation type (e.g. the gradient of gravitational potential where gravitational potential is the term and gradient is the operation).

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::forms

Concepts

- concept serif::discretization::forms::IsOperation
- concept serif::discretization::forms::IsOperand

  Concept to validate a valid operand. An operand is valid if the following conditions are satisfied.

11.73.1 Detailed Description

This file defines the Operand type, which represents a term and an operation to be applied to that term in the context of a form.

Author

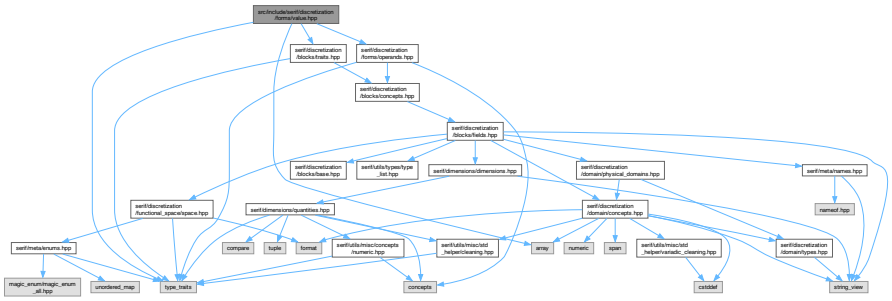
Emily M. Boudreaux

An Operand is a compile-time type that encapsulates a term and an operation to be applied to that term. The operation can be one of several predefined operations, such as value, gradient, divergence, curl, or normal trace. The Operand type is used in the definition of forms to specify the terms and operations involved in the form.

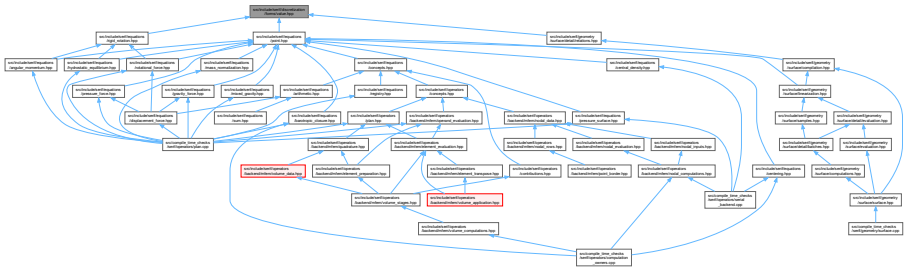
11.74 src/include/serif/discretization/forms/value.hpp File Reference

Encode compile time information about the shape (rank) and data type of the result of various Operations.

#include <array>
#include <type_traits>
#include "serif/discretization/blocks/traits.hpp"
#include "serif/discretization/forms/operands.hpp"
Include dependency graph for value.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::forms::OperationValue< 0, Operation::Value >

  The value of a rank 0 tensor is a scalar, a double in the case of SERiF where we restrict ourselves to double precision.
- struct serif::discretization::forms::OperationValue< 1, Operation::Value >

  The value of a rank 1 tensor is a rank 1 tensor (a vector of doubles in SERiF)
- struct serif::discretization::forms::OperationValue< 1, Operation::Divergence >

  The divergence of a rank one tensor is a scalar (a double)
- struct serif::discretization::forms::OperationValue< 0, Operation::Gradient >

  The gradient of a rank 0 tensor is a rank 1 tensor (a vector of doubles)
- struct serif::discretization::forms::OperationValue< 1, Operation::Curl >

  The curl of a rank 1 tensor is a rank 1 tensor (a vector of doubles)
- struct serif::discretization::forms::OperationValue< 1, Operation::NormalTrace >
- struct serif::discretization::forms::OperandResult< OperandT >

  Specialization of the OperandResult struct for operands that are valid operands (satisfy the IsOperand concept) and are spatial terms (e.g defined over the domain, think mass density). This specialization uses the OperationValue struct to determine the value type of the result of the operand based on its term type and operation type.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::forms

Typedefs

- using serif::discretization::forms::VectorValue = std::array<double, 3>

  Alias for a rank 1 tensor of dimension 3 with elements doubles.
- using serif::discretization::forms::MatrixValue = std::array<VectorValue, 3>

  Alias for a rank 2 tensor of dimension 3 with cells as doubles.
- template<IsOperand OperandT>
  using serif::discretization::forms::OperandValue = OperandResult<OperandT>::type

Alias for the value type of an operand. This is a convenience alias that allows you to easily access the value type of an operand without having to explicitly use the OperandResult struct.

- template<typename OutputT, typename InputT>
  using serif::discretization::forms::PartialDerivative = std::conditional_t<std::same_as<OutputT, double>, InputT, std::conditional_t<std::same_as<InputT, double>, OutputT, MatrixValue>>

  Alias for the partial derivative of some output type with respect to some input type. This is a convenience alias that allows you to easily access the partial derivative type without having to explicitly use the OperandResult struct.

Functions

- template<typename ValueT>
  decltype(auto) serif::discretization::forms::component (ValueT &value, int index)

  Read one entry of a scalar/vector/matrix value. This is a convenience function that allows you to easily access the components of a value type without having to explicitly use the OperandResult struct.
- template<typename OutputT, typename InputT>
  double serif::discretization::forms::derivative_component (const PartialDerivative< OutputT, InputT > &value, int output, int input)

  Read one entry of a scalar/vector/matrix local derivative.

Variables

- template<typename ValueT>
  constexpr int serif::discretization::forms::component_count = std::same_as<ValueT, double> ? 1 : 3

  Helper variable template to determine the number of components in the value type of an operand. This is a convenience variable that allows you to easily access the number of components in the value type of an operand without having to explicitly use the OperandResult struct.

11.74.1 Detailed Description

Encode compile time information about the shape (rank) and data type of the result of various Operations.

Author

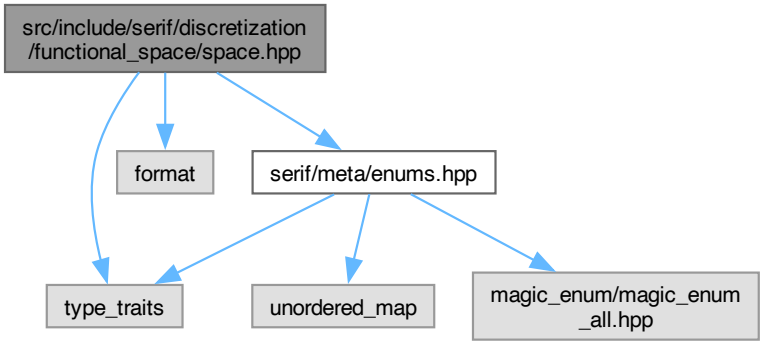
Emily M. Boudreaux

Todo This should be consolidated and removed. This information could be injected into the Operation itself.

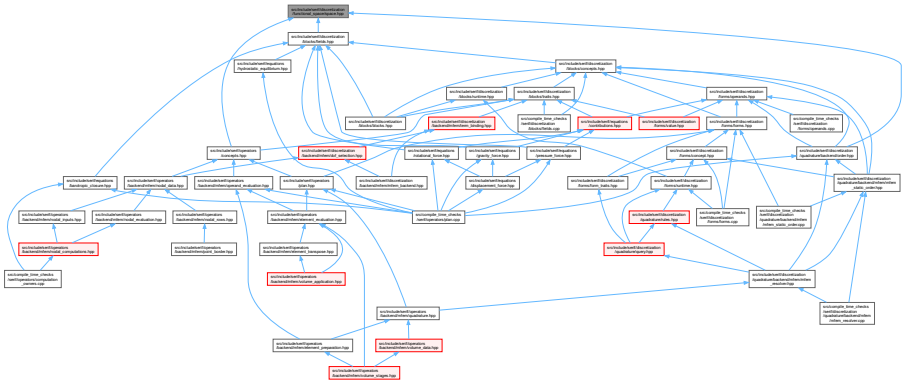
11.75 src/include/serif/discretization/functional_space/space.hpp File Reference

This file defines the functional space types used in the discretization framework, including compile-time descriptors and runtime descriptors.

#include <type_traits>
#include <format>
#include "serif/meta/enums.hpp"
Include dependency graph for space.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::functional_space::Rank< R >

  Compile-time descriptor for the rank of a functional space.
- struct serif::discretization::functional_space::SpaceDescriptor

Compile-time descriptor for a functional space. This struct is used to describe the family, continuity, trace, and mapping of a functional space at compile time.

- struct serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >

  Compile-time descriptor for a discrete functional space. The difference between a FunctionalSpace and a discrete functional space is that a discrete functional space carries some additional information which is needed to actually construct a finite element space (e.g. polynomial order, and rank). Once you have a discrete functional space you have, for MFEM at least, enough information to fully define a finite element space.
- struct std::formatter< serif::discretization::functional_space::Rank< R > >

  Specialization of the std::formatter for the functional space types. This allows us to use std::format to format the functional space types as strings.
- struct std::formatter< serif::discretization::functional_space::SpaceDescriptor >

  Specialization of the std::formatter for the SpaceDescriptor types. This allows us to use std::format to format the SpaceDescriptor types as strings.
- struct std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > >

  Specialization of the std::formatter for the DiscreteFunctionalSpace types. This allows us to use std::format to format the DiscreteFunctionalSpace types as strings.
- struct std::formatter< serif::discretization::functional_space::Family >

  Specialization of the std::formatter for the Family enum. This allows us to use std::format to format the Family enum as strings.
- struct std::formatter< serif::discretization::functional_space::Continuity >

  Specialization of the std::formatter for the Continuity enum. This allows us to use std::format to format the Continuity enum as strings.
- struct std::formatter< serif::discretization::functional_space::Trace >

  Specialization of the std::formatter for the Trace enum. This allows us to use std::format to format the Trace enum as strings.
- struct std::formatter< serif::discretization::functional_space::Mapping >

  Specialization of the std::formatter for the Mapping enum. This allows us to use std::format to format the Mapping enum as strings.

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::functional_space

Concepts

- concept serif::discretization::functional_space::IsSpaceDescriptor

  Concept to validate a valid space descriptor. A space descriptor is valid if it is of type SpaceDescriptor. This is a very simple concept that just checks that the type is exactly SpaceDescriptor.

Enumerations

- enum class serif::discretization::functional_space::Family { serif::discretization::functional_space::l2 , serif::discretization::functional_space::h1 , serif::discretization::functional_space::hdiv , serif::discretization::functional_ }

  Compile-time descriptor for the family, continuity, trace, and mapping of a functional space.
- enum class serif::discretization::functional_space::Continuity { serif::discretization::functional_space::discontinuous , serif::discretization::functional_space::continuous , serif::discretization::functional_space::normal_continuous , serif::discretization::functional_space::tangential_continuous }

  Compile-time descriptor for the continuity of a functional space. This enum is parallel to the Family enum.
- enum class serif::discretization::functional_space::Trace { serif::discretization::functional_space::none , serif::discretization::functional_space::value , serif::discretization::functional_space::normal , serif::discretization::functional_space::tangential }

  Compile-time descriptor for the trace of a functional space. This enum is parallel to the Family enum.
- enum class serif::discretization::functional_space::Mapping { serif::discretization::functional_space::identity , serif::discretization::functional_space::contravariant_piola , serif::discretization::functional_space::covariant_piola }

  Compile-time descriptor for the mapping of a functional space. This tells the DomainMapper which piola transformations to use when moving between reference and physical spaces.

Variables

- static auto serif::discretization::functional_space::Family2StringMap = meta::enum_to_enum_string_map<Family>()

  Mapping from Family enum values to their string representations.
- static auto serif::discretization::functional_space::String2FamilyMap = meta::enum_to_string_enum_map<Family>()

  Mapping from string representations to Family enum values.
- static auto serif::discretization::functional_space::Continuity2StringMap = meta::enum_to_enum_string_map<Contin

  Mapping from Continuity enum values to their string representations.
- static auto serif::discretization::functional_space::String2ContinuityMap = meta::enum_to_string_enum_map<Contin

  Mapping from string representations to Continuity enum values.
- static auto serif::discretization::functional_space::Trace2StringMap = meta::enum_to_enum_string_map<Trace>()

  Mapping from Trace enum values to their string representations.
- static auto serif::discretization::functional_space::String2TraceMap = meta::enum_to_string_enum_map<Trace>()

  Mapping from string representations to Trace enum values.
- static auto serif::discretization::functional_space::Mapping2StringMap = meta::enum_to_enum_string_map<Mapping

  Mapping from Mapping enum values to their string representations.
- static auto serif::discretization::functional_space::String2MappingMap = meta::enum_to_string_enum_map<Mapping

  Mapping from string representations to Mapping enum values.
- static constexpr SpaceDescriptor serif::discretization::functional_space::L2

  Compile-time descriptor for an L2 functional space. This is a convenience descriptor that allows you to easily create an L2 functional space descriptor without having to specify all the parameters. The specifications are.
- static constexpr SpaceDescriptor serif::discretization::functional_space::H1

  Compile-time descriptor for an H1 functional space. This is a convenience descriptor that allows you to easily create an H1 functional space descriptor without having to specify all the parameters. The specifications are.
- static constexpr SpaceDescriptor serif::discretization::functional_space::Hdiv

  Compile-time descriptor for an H(div) functional space. This is a convenience descriptor that allows you to easily create an H(div) functional space descriptor without having to specify all the parameters. The specifications are.

- static constexpr SpaceDescriptor serif::discretization::functional_space::Hcurl

  Compile-time descriptor for an H(curl) functional space. This is a convenience descriptor that allows you to easily create an H(curl) functional space descriptor without having to specify all the parameters. The specifications are.

11.75.1 Detailed Description

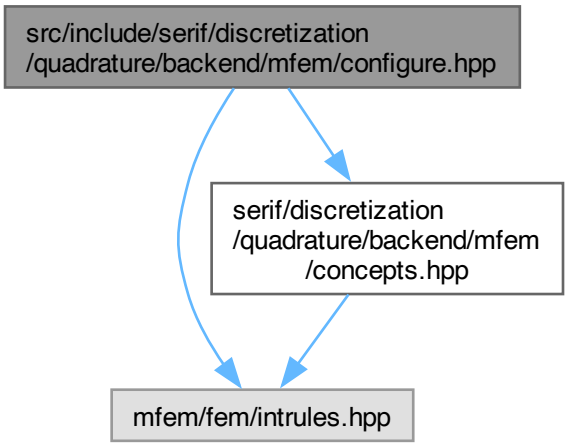
This file defines the functional space types used in the discretization framework, including compile-time descriptors and runtime descriptors.

Author

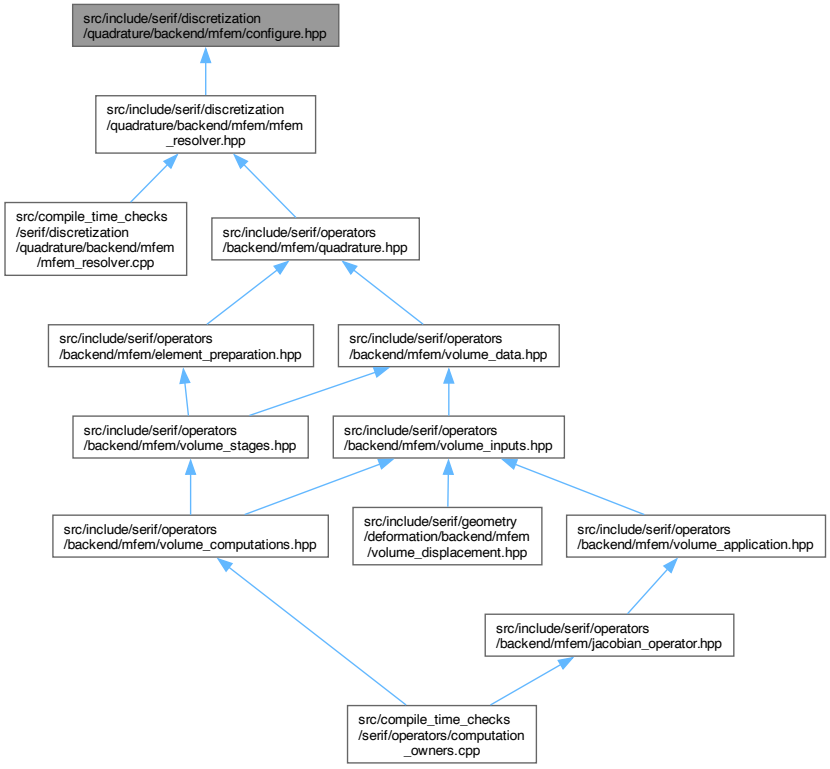
Emily M. Boudreaux

11.76 src/include/serif/discretization/quadrature/backend/mfem/configure.hpp File Reference

#include "mfem/fem/intrules.hpp"
#include "serif/discretization/quadrature/backend/mfem/concepts.hpp"
Include dependency graph for configure.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::quadrature
- namespace serif::discretization::quadrature::backend
- namespace serif::discretization::quadrature::backend::mfem

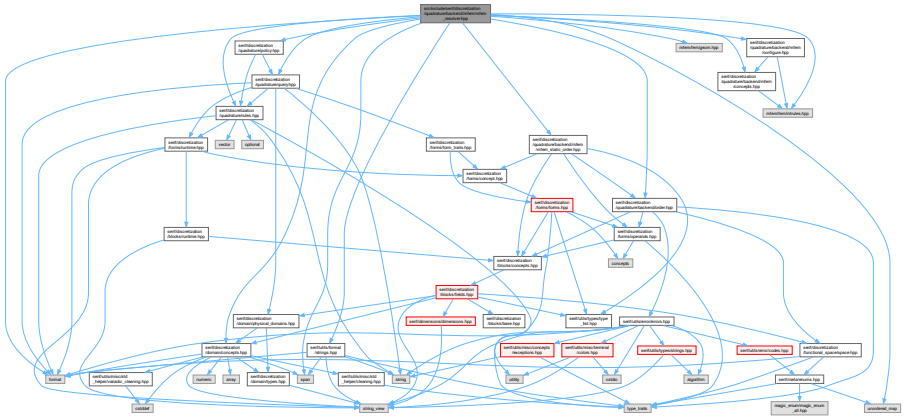
Functions

- template<ConfigurableIntegrator IntegratorT>
  void serif::discretization::quadrature::backend::mfem::set_integration_rule (IntegratorT &integrator, const ::mfem::IntegrationRule &integration_rule)

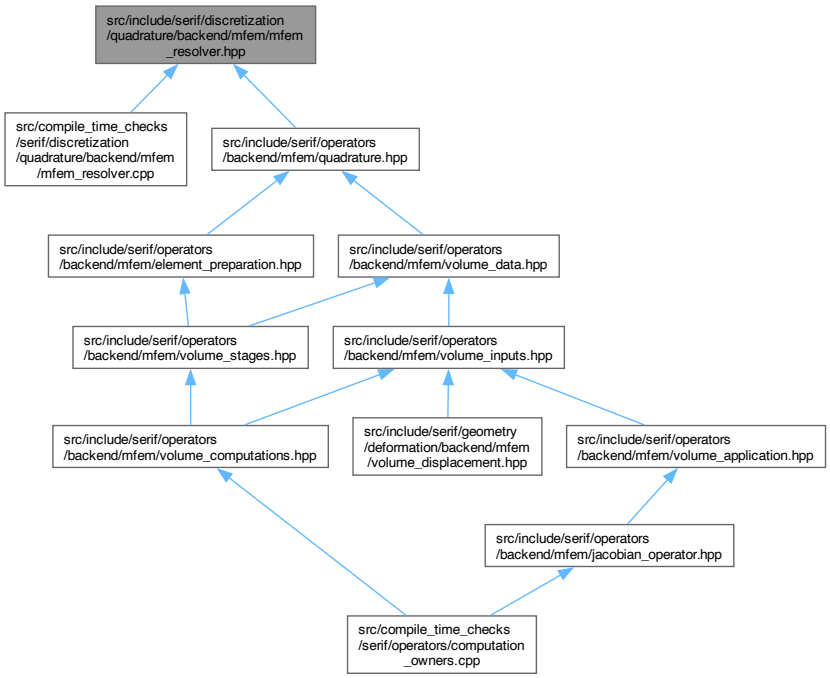
11.77 src/include/serif/discretization/quadrature/backend/mfem/mfem_↩ resolver.hpp File Reference

#include
#include <format>

#include <unordered_map>
#include "mfem/fem/geom.hpp"
#include "mfem/fem/intrules.hpp"
#include "serif/discretization/quadrature/rules.hpp"
#include "serif/discretization/quadrature/backend/order.hpp"
#include "serif/discretization/domain/concepts.hpp"
#include "serif/discretization/quadrature/policy.hpp"
#include "serif/discretization/quadrature/query.hpp"
#include "serif/discretization/quadrature/backend/mfem/concepts.hpp"
#include "serif/discretization/quadrature/backend/mfem/configure.hpp"
#include "serif/discretization/quadrature/backend/mfem/mfem_static_order.hpp"
#include "serif/utils/format/strings.hpp"
Include dependency graph for mfem_resolver.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext
- struct serif::discretization::quadrature::backend::mfem::MFEMRule
- class serif::discretization::quadrature::backend::mfem::MFEMResolver
- struct std::formatter< serif::discretization::quadrature::backend::mfem::MFEMQuadratureContext >

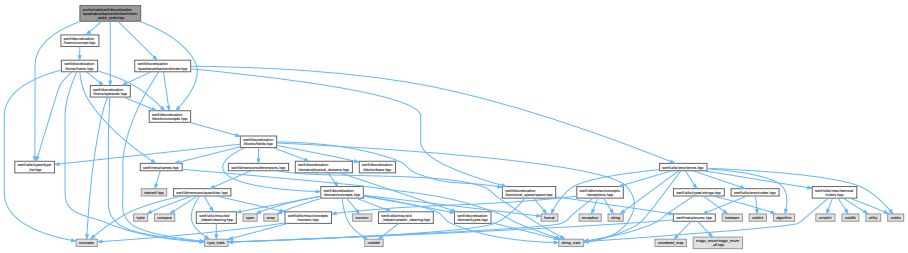
Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::quadrature
- namespace serif::discretization::quadrature::backend
- namespace serif::discretization::quadrature::backend::mfem

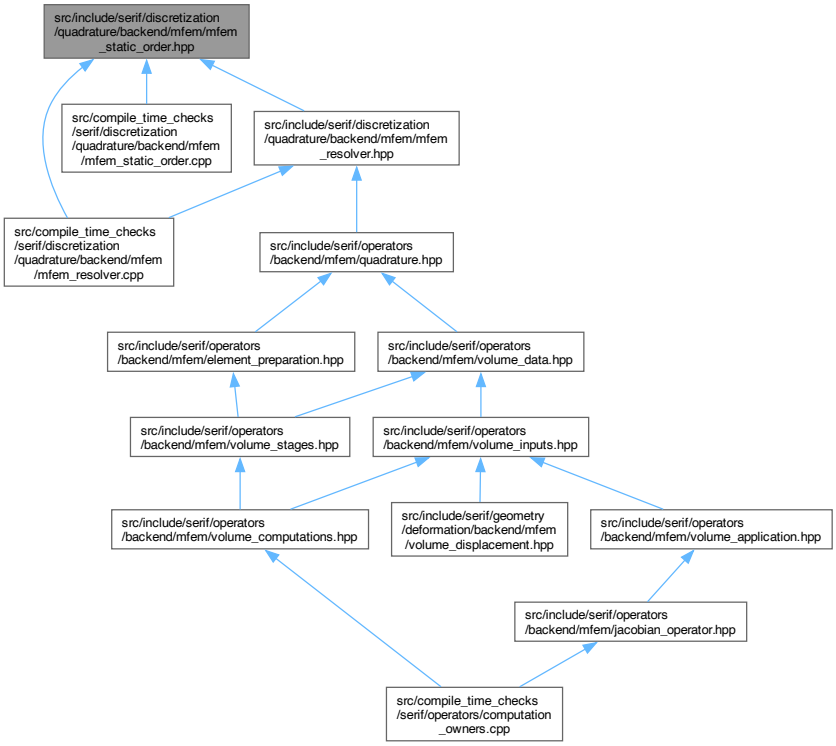
11.78 src/include/serif/discretization/quadrature/backend/mfem/mfem_↩ static_order.hpp File Reference

#include "serif/discretization/blocks/concepts.hpp"
#include "serif/discretization/forms/operands.hpp"

#include ”serif/discretization/forms/concept.hpp”
#include ”serif/utils/types/type_list.hpp”
#include ”serif/discretization/quadrature/backend/order.hpp”
Include dependency graph for mfem_static_order.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::quadrature::backend::mfem::MFEMOperandListOrder< utils::types::TypeList< OperandTs.

Namespaces

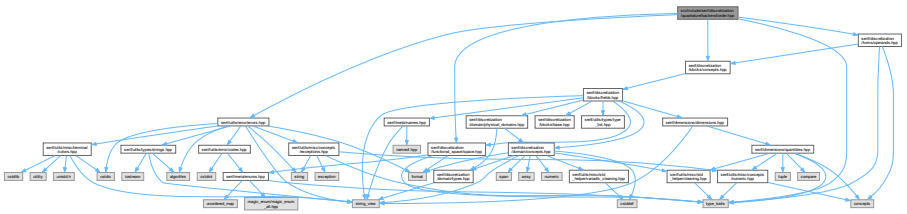
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::quadrature
- namespace serif::discretization::quadrature::backend
- namespace serif::discretization::quadrature::backend::mfem

Variables

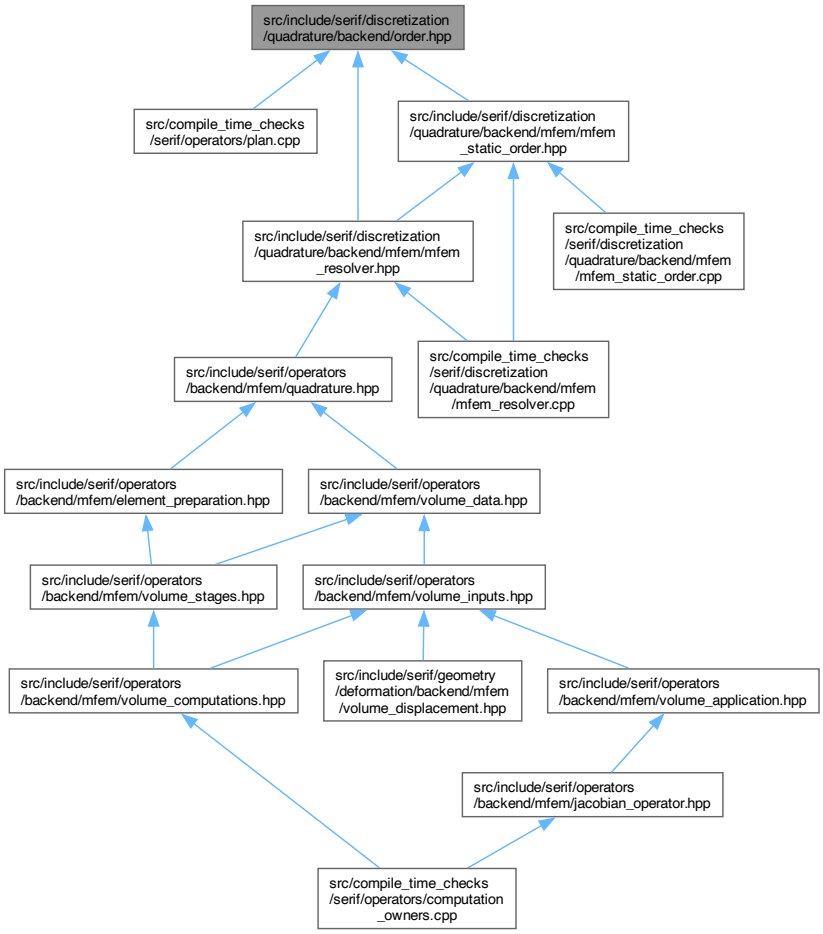
- template<forms::Form FormT>
  constexpr int serif::discretization::quadrature::backend::mfem::mfem_form_static_order = MFEMOperandListOrder<t
  FormT::operand_types>::value

11.79 src/include/serif/discretization/quadrature/backend/order.hpp File Reference

#include <type_traits>
#include "serif/discretization/functional_space/space.hpp"
#include "serif/discretization/blocks/concepts.hpp"
#include "serif/discretization/forms/operands.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for order.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSpaceTraits< functional_space::DiscreteFunc

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::quadrature
- namespace serif::discretization::quadrature::backend
- namespace serif::discretization::quadrature::backend::mfem

Typedefs

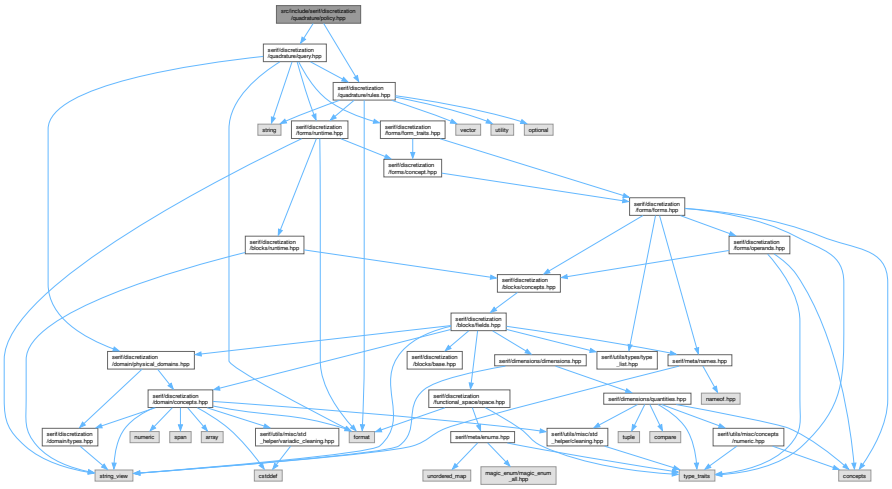
- template<blocks::HasFunctionalSpace TermT>
  using serif::discretization::quadrature::backend::mfem::FunctionalSpaceTraits = DiscreteFunctionalSpaceTraits<typenar
  TermT::FunctionalSpace>

Functions

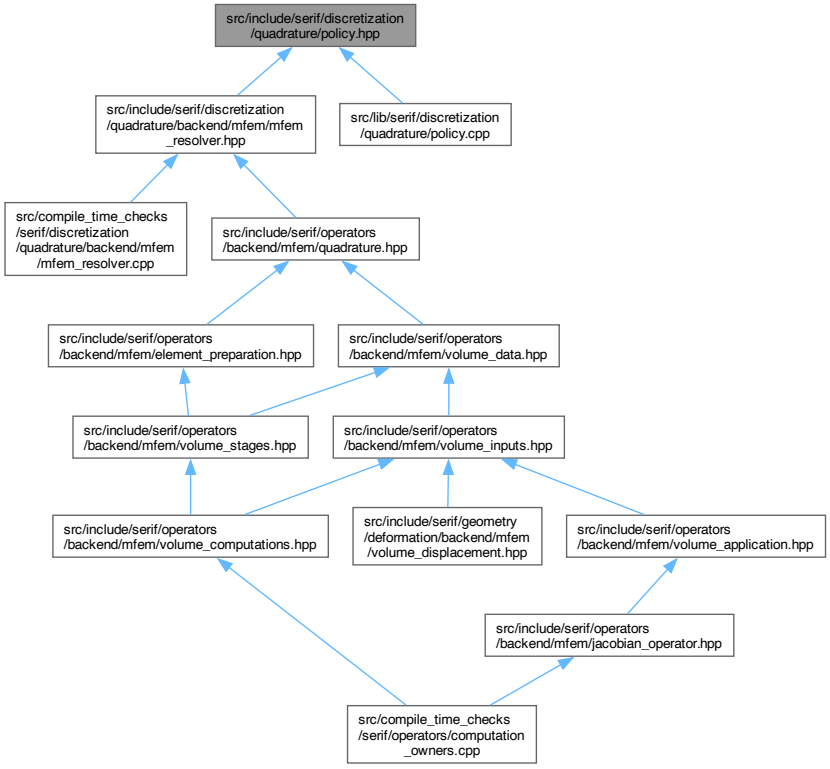
- template<forms::IsOperand OperandT>
  consteval int serif::discretization::quadrature::backend::mfem::resolve_operand_order () noexcept
- int serif::discretization::quadrature::backend::mfem::sum_dynamic_orders (const std::span< const int > dynamic_orders)

11.80 src/include/serif/discretization/quadrature/policy.hpp File Reference

#include "serif/discretization/quadrature/query.hpp"
#include "serif/discretization/quadrature/rules.hpp"
Include dependency graph for policy.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::discretization::quadrature::Policy

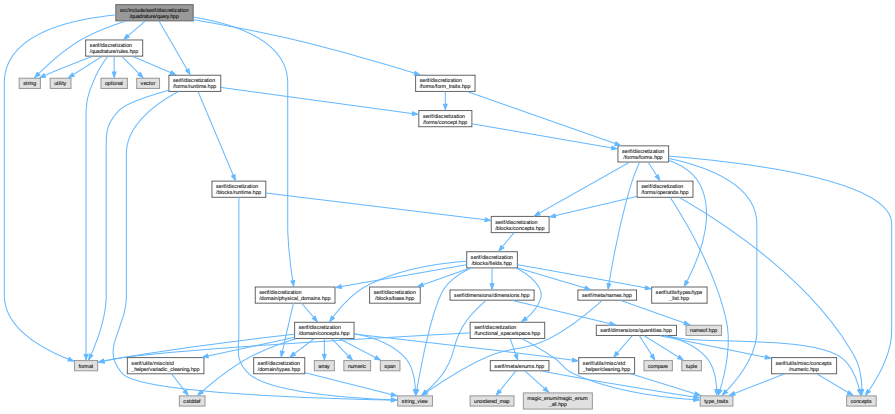
Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::quadrature

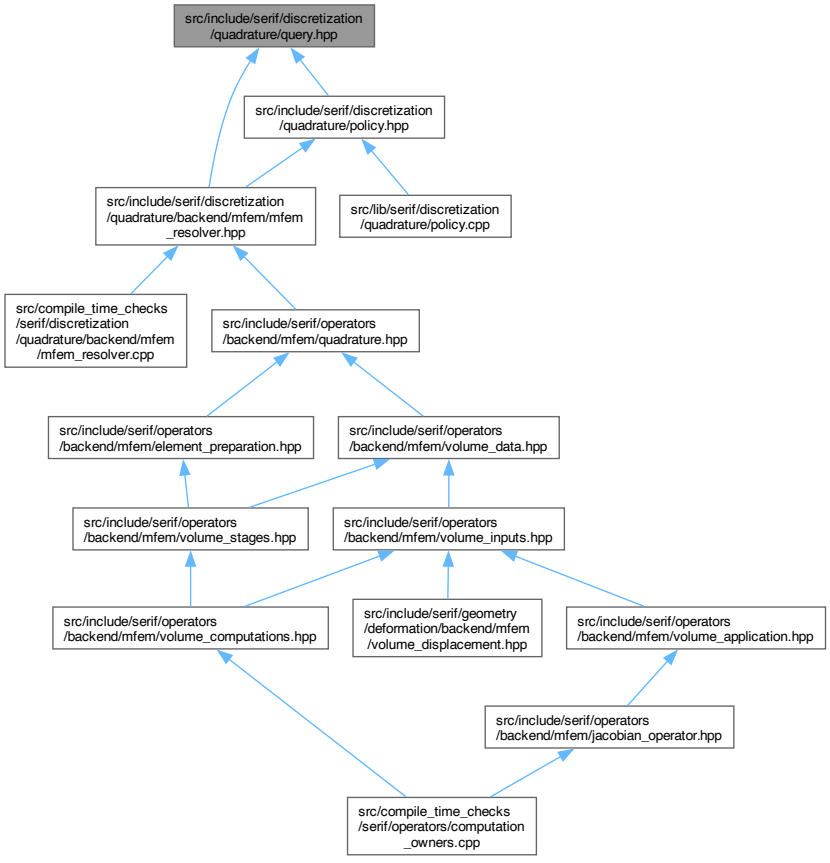
11.81 src/include/serif/discretization/quadrature/query.hpp File Reference

#include <format>
#include <string>
#include "serif/discretization/quadrature/rules.hpp"
#include "serif/discretization/domain/physical_domains.hpp"
#include "serif/discretization/forms/form_traits.hpp"

#include "serif/discretization/forms/runtime.hpp"
Include dependency graph for query.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::quadrature::Query
- struct std::formatter< serif::discretization::quadrature::Query >

Namespaces

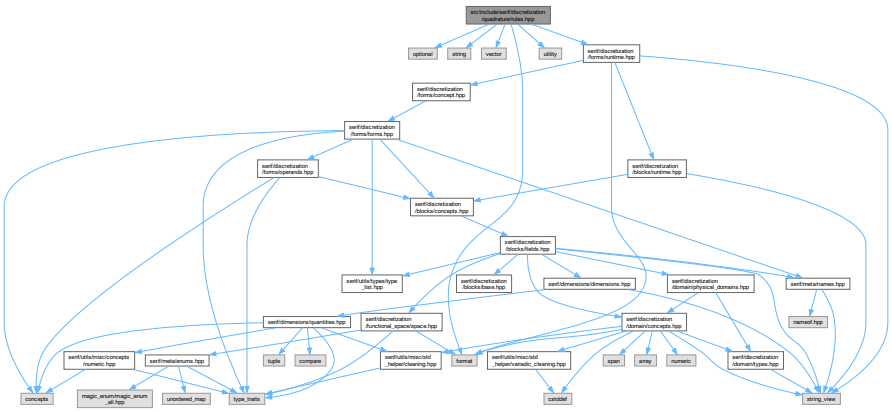
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::quadrature

Functions

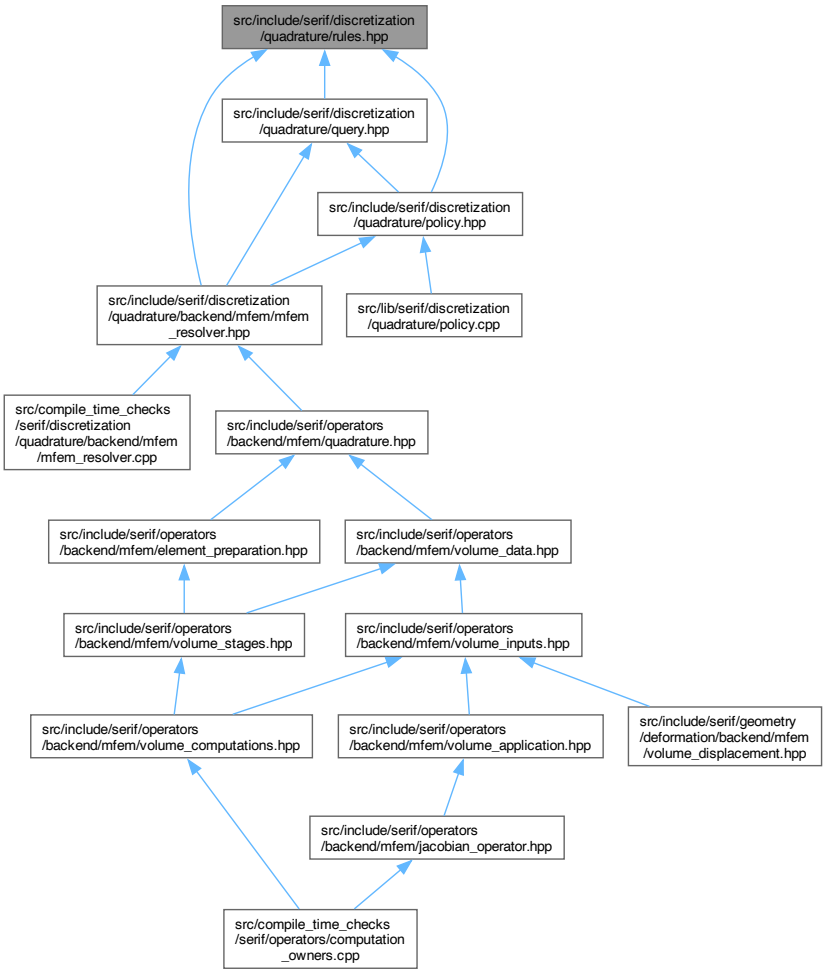
- template<forms::Form FormT>
  constexpr Query serif::discretization::quadrature::make_query (const QuadratureRole role, const domain::DomainOrSet domain, const MappingKind mapping, const int trialOrder, const int testOrder, const int coefficientOrder, const int geometryWeightOrder, const std::optional< int > baseOrder=std::nullopt) noexcept

11.82 src/include/serif/discretization/quadrature/rules.hpp File Reference

#include <optional>
#include <string>
#include <vector>
#include <format>
#include <utility>
#include "serif/discretization/forms/runtime.hpp"
Include dependency graph for rules.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::quadrature::RuleControl
- struct serif::discretization::quadrature::FormRule
- class serif::discretization::quadrature::RuleSet
- struct serif::discretization::quadrature::RoleControls
- struct serif::discretization::quadrature::Resolution
- struct serif::discretization::quadrature::QuadratureTermOptions
- struct serif::discretization::quadrature::QuadratureManifestOptions
- struct serif::discretization::quadrature::QuadratureValidationOptions
- struct serif::discretization::quadrature::QuadratureRoleOptions
- struct serif::discretization::quadrature::QuadratureOptions
- struct std::formatter< serif::discretization::quadrature::QuadratureRole >
- struct std::formatter< serif::discretization::quadrature::MappingKind >

- struct std::formatter< serif::discretization::quadrature::Mode >
- struct std::formatter< serif::discretization::quadrature::RuleControl >
- struct std::formatter< serif::discretization::quadrature::FormRule >
- struct std::formatter< serif::discretization::quadrature::RuleSet >

Namespaces

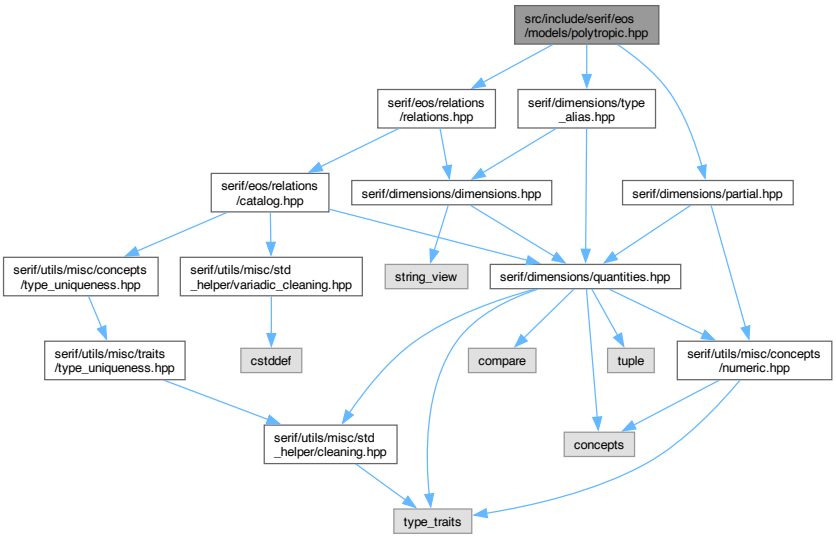
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::quadrature

Enumerations

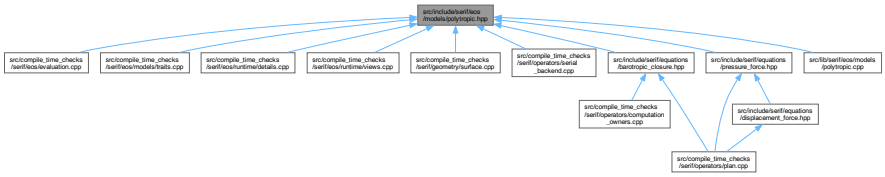
- enum class serif::discretization::quadrature::QuadratureRole { serif::discretization::quadrature::discretization , serif::discretization::quadrature::preconditioner , serif::discretization::quadrature::diagnostic , serif::discretization::quadrature::projection }
- enum class serif::discretization::quadrature::MappingKind { serif::discretization::quadrature::none , serif::discretization::quadrature::affine , serif::discretization::quadrature::general , serif::discretization::quadrature::kelvin }
- enum class serif::discretization::quadrature::Mode { serif::discretization::quadrature::fast , serif::discretization::quadratur , serif::discretization::quadrature::reference , serif::discretization::quadrature::convergence }

11.83 src/include/serif/eos/models/polytropic.hpp File Reference

#include "serif/eos/relations/relations.hpp"
#include "serif/dimensions/type_alias.hpp"
#include "serif/dimensions/partial.hpp"
Include dependency graph for polytropic.hpp:



This graph shows which files directly or indirectly include this file:



Classes

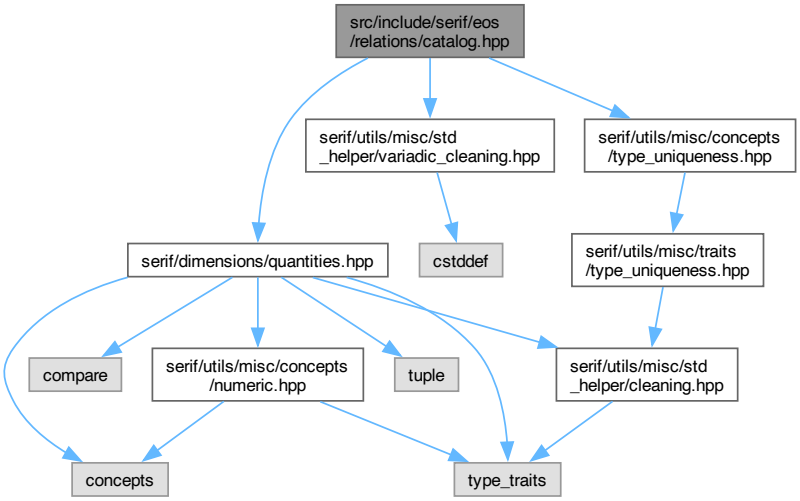
- struct serif::eos::models::PolytropeParameters
- class serif::eos::models::Polytrope

Namespaces

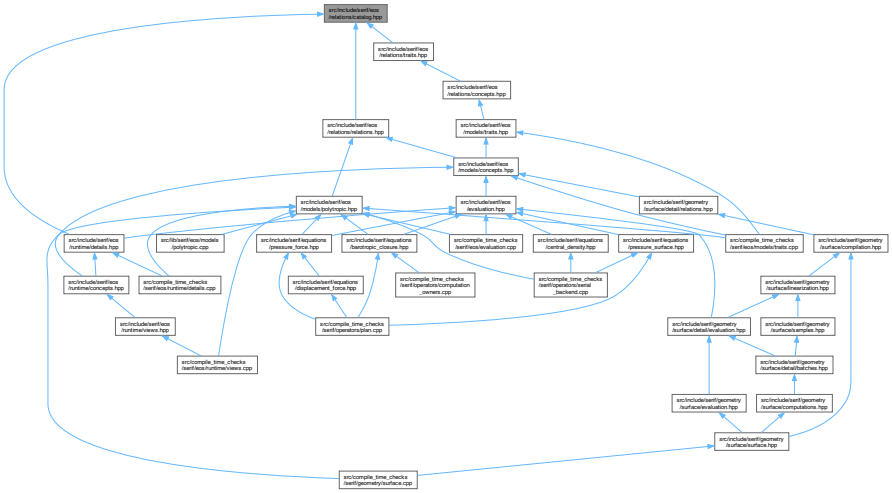
- namespace serif
- namespace serif::eos
- namespace serif::eos::models

11.84 src/include/serif/eos/relations/catalog.hpp File Reference

#include "serif/utils/misc/concepts/type_uniqueness.hpp"
#include "serif/utils/misc/std_helper/variadic_cleaning.hpp"
#include "serif/dimensions/quantities.hpp"
Include dependency graph for catalog.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::eos::relations::Relation< Output, Inputs >
- struct serif::eos::relations::RelationCatalog< Relations >

Namespaces

- namespace serif
- namespace serif::eos
- namespace serif::eos::relations

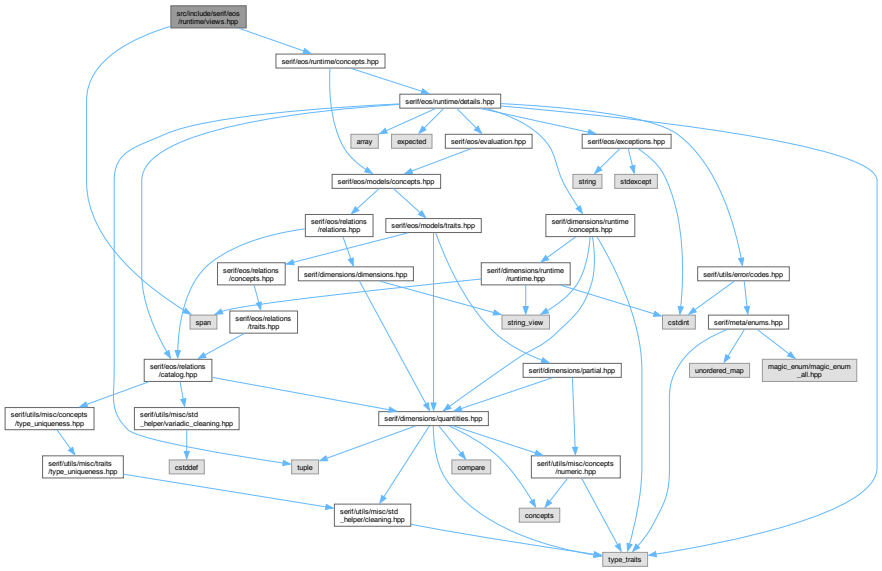
Functions

- template<typename... Ts>
  consteval std::size_t serif::eos::relations::CountVariadicArguments () noexcept

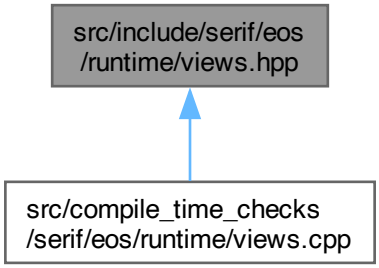
11.85 src/include/serif/eos/runtime/views.hpp File Reference

#include
#include "serif/eos/runtime/concepts.hpp"

Include dependency graph for views.hpp:



This graph shows which files directly or indirectly include this file:



Classes

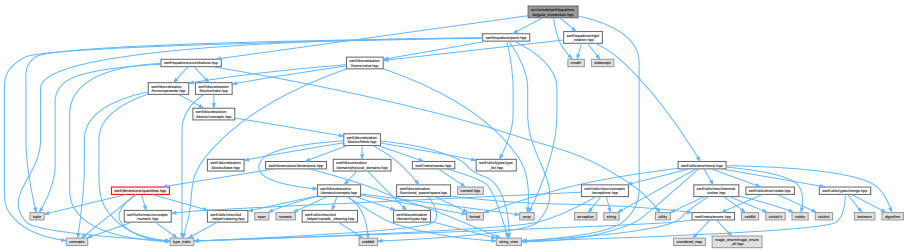
- class serif::eos::runtime::EOSView

Namespaces

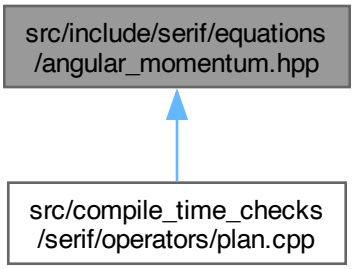
- namespace serif
- namespace serif::eos
- namespace serif::eos::runtime

11.86 src/include/serif/equations/angular_momentum.hpp File Reference

#include <cmath>
#include <string_view>
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"
#include "serif/equations/rigid_rotation.hpp"
Include dependency graph for angular_momentum.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::equations::AngularMomentum

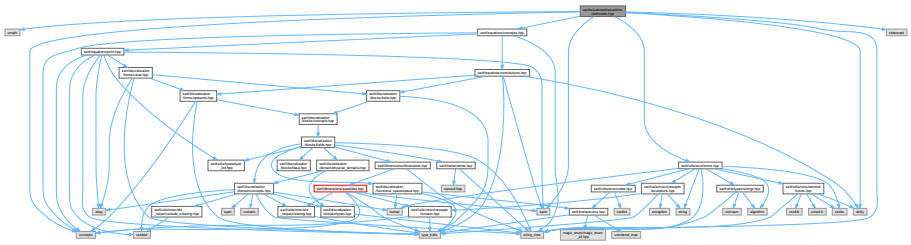
  Close the unknown angular speed with integral_star rho Omega s^2 dV - target_momentum = 0.

Namespaces

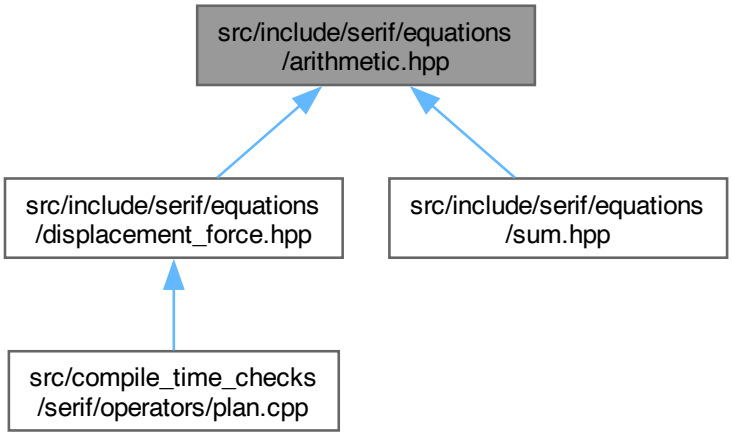
- namespace serif
- namespace serif::equations

11.87 src/include/serif/equations/arithmetic.hpp File Reference

#include <cmath>
#include <concepts>
#include <stdexcept>
#include <string_view>
#include <tuple>
#include <utility>
#include "serif/equations/concepts.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for arithmetic.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::equations::EquationScale< EquationT >

  Multiply an equation's weak contributions by a constant independent of state and position.
- class serif::equations::EquationSum< FirstT, RestTs >

Add equations sharing a residual row; each retains its own test operation and support.

- class serif::equations::EquationDifference< LeftT, RightT >

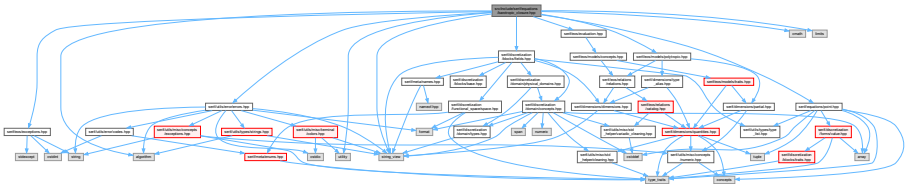
  Subtract every weak contribution of the right equation from the left equation.

Namespaces

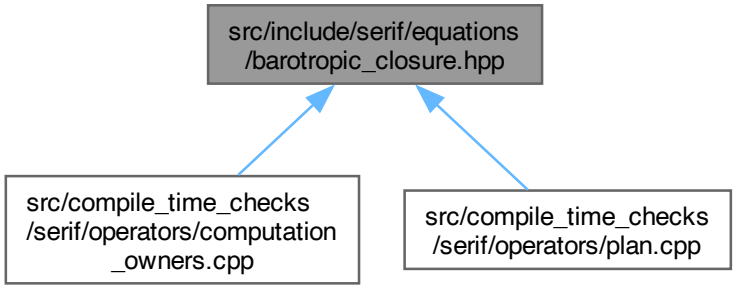
- namespace serif
- namespace serif::equations

11.88 src/include/serif/equations/barotropic_closure.hpp File Reference

#include <algorithm>
#include <cmath>
#include <limits>
#include <string_view>
#include <utility>
#include "serif/discretization/blocks/fields.hpp"
#include "serif/eos/evaluation.hpp"
#include "serif/eos/exceptions.hpp"
#include "serif/eos/models/polytropic.hpp"
#include "serif/equations/point.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for barotropic_closure.hpp:



This graph shows which files directly or indirectly include this file:



Classes

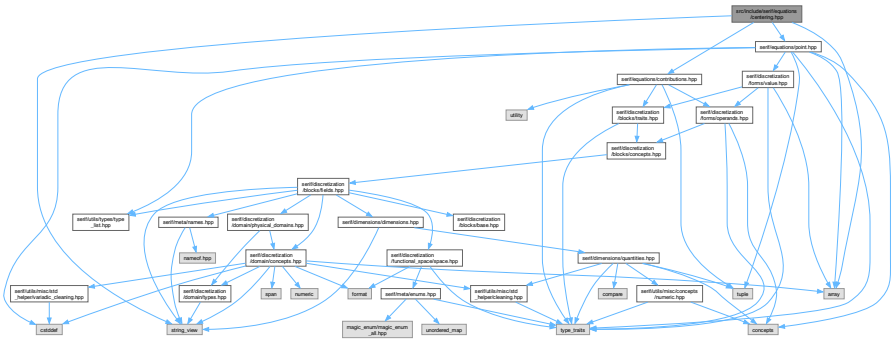
- class serif::equations::BarotropicClosure< EOST >

Namespaces

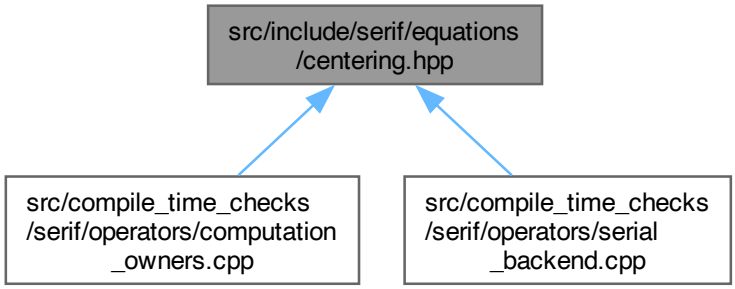
- namespace serif
- namespace serif::equations

11.89 src/include/serif/equations/centering.hpp File Reference

#include <array>
#include <string_view>
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"
Include dependency graph for centering.hpp:



This graph shows which files directly or indirectly include this file:



Classes

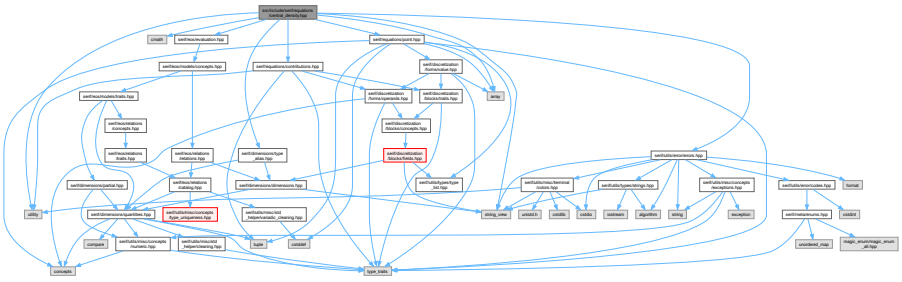
- struct serif::equations::CenteringConstraint

Namespaces

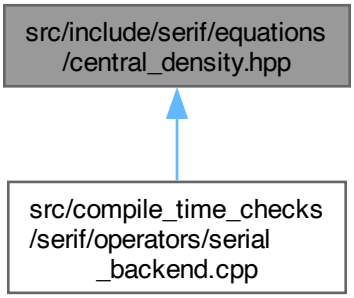
- namespace serif
- namespace serif::equations

11.90 src/include/serif/equations/central_density.hpp File Reference

#include <array>
#include <cmath>
#include <string_view>
#include <utility>
#include "serif/dimensions/type_alias.hpp"
#include "serif/eos/evaluation.hpp"
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for central_density.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::equations::CentralDensityConstraint< EOST >

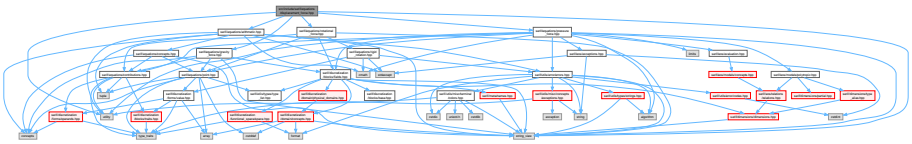
  Prescribe central density through the continuous enthalpy: h(0) - h_EOS(rho_c) = 0.

Namespaces

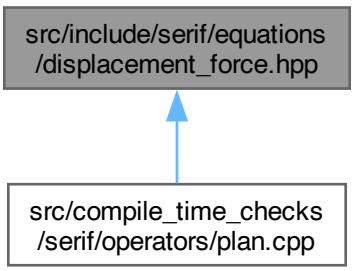
- namespace serif
- namespace serif::equations

11.91 src/include/serif/equations/displacement_force.hpp File Reference

#include <string_view>
#include <utility>
#include "serif/equations/arithmetic.hpp"
#include "serif/equations/gravity_force.hpp"
#include "serif/equations/pressure_force.hpp"
#include "serif/equations/rotational_force.hpp"
Include dependency graph for displacement_force.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::equations::DisplacementForce< EOST >

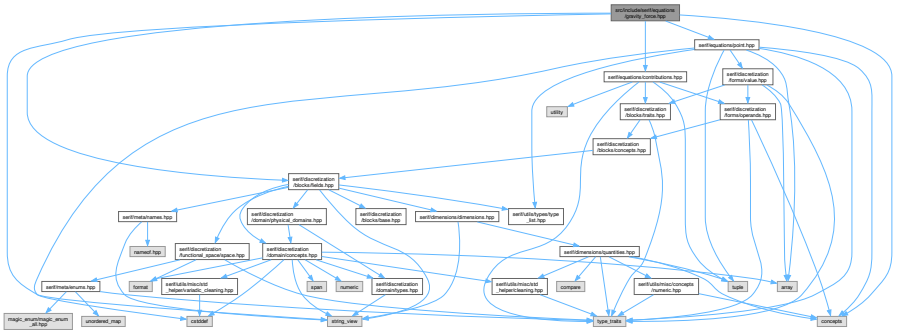
  Mechanical balance: pressure force + gravity force + rotational force.

Namespaces

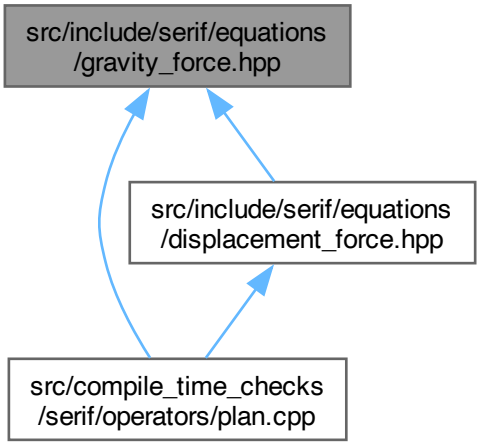
- namespace serif
- namespace serif::equations

11.92 src/include/serif/equations/gravity_force.hpp File Reference

#include <concepts>
#include <string_view>
#include "serif/discretization/blocks/fields.hpp"
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"
Include dependency graph for gravity_force.hpp:



This graph shows which files directly or indirectly include this file:



Classes

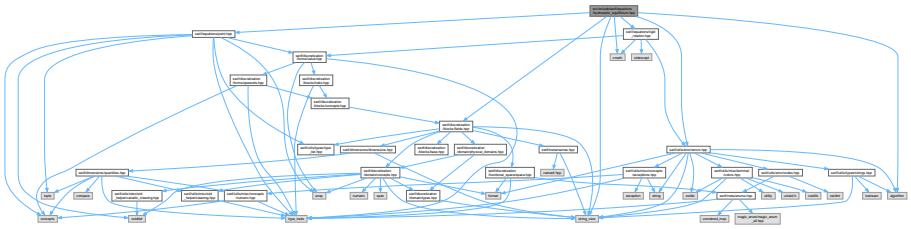
- struct serif::equations::GravityForce

Namespaces

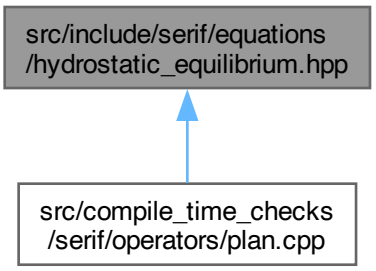
- namespace serif
- namespace serif::equations

11.93 src/include/serif/equations/hydrostatic_equilibrium.hpp File Reference

#include <algorithm>
#include <cmath>
#include <string_view>
#include "serif/discretization/blocks/fields.hpp"
#include "serif/equations/point.hpp"
#include "serif/equations/rigid_rotation.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for hydrostatic_equilibrium.hpp:



This graph shows which files directly or indirectly include this file:



Classes

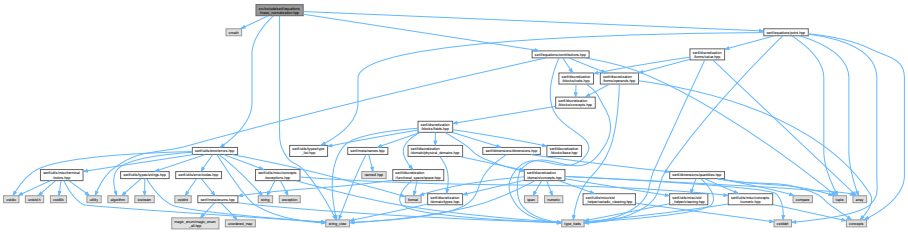
- class serif::equations::BarotropicHydrostaticEquilibrium

Namespaces

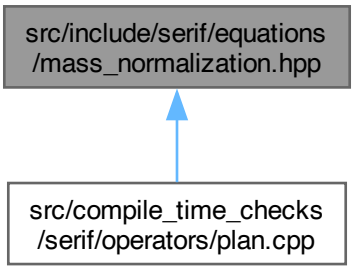
- namespace serif
- namespace serif::equations

11.94 src/include/serif/equations/mass_normalization.hpp File Reference

#include <cmath>
#include <string_view>
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for mass_normalization.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::equations::MassNormalization

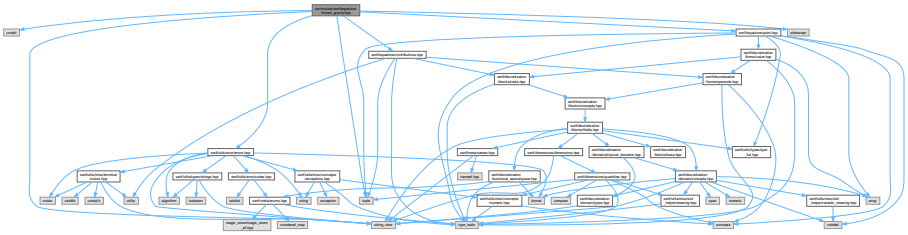
  Close the unknown Bernoulli constant with integral_star rho dV - target_mass = 0.

Namespaces

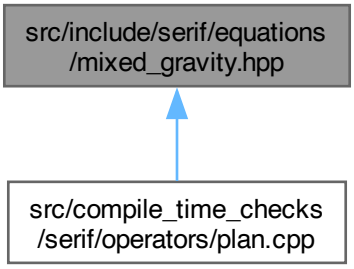
- namespace serif
- namespace serif::equations

11.95 src/include/serif/equations/mixed_gravity.hpp File Reference

#include <cmath>
#include <tuple>
#include <string_view>
#include <stdexcept>
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for mixed_gravity.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::equations::gravity_contributions::GradientValue
- struct serif::equations::gravity_contributions::PotentialValue
- struct serif::equations::gravity_contributions::GradientDivergence
- struct serif::equations::gravity_contributions::DensitySource
- struct serif::equations::GravityGradientRelation
- class serif::equations::PoissonBalance

Namespaces

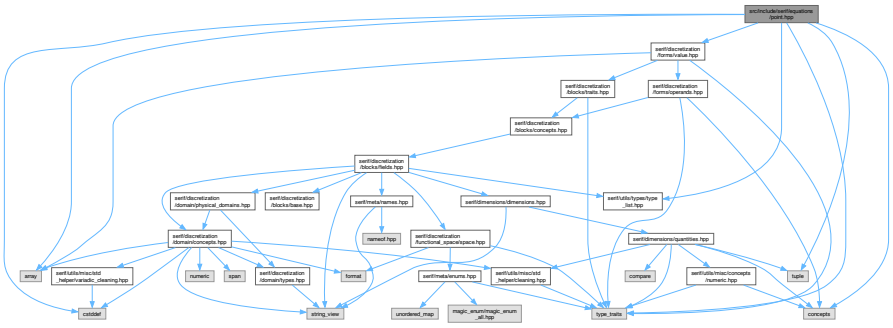
- namespace serif
- namespace serif::equations
- namespace serif::equations::gravity_contributions

Typedefs

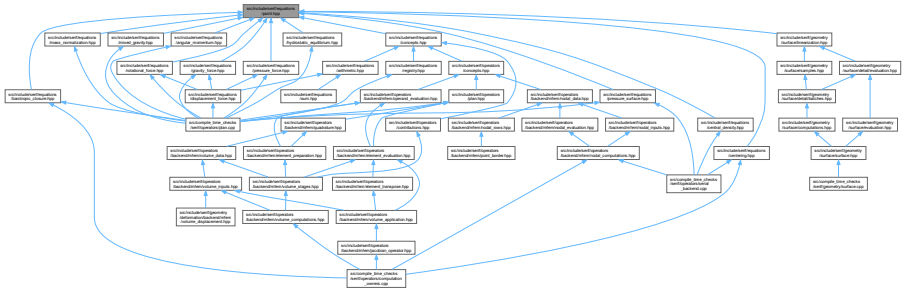
- using serif::equations::gravity_contributions::Gradient = discretization::blocks::gravity::gradient
- using serif::equations::gravity_contributions::Potential = discretization::blocks::gravity::potential
- using serif::equations::gravity_contributions::Density = discretization::blocks::density::mass
- using serif::equations::gravity_contributions::Value = discretization::forms::Operation::Value
- using serif::equations::gravity_contributions::Divergence = discretization::forms::Operation::Divergence
- template<typename Term, typename Operation = Value>
  using serif::equations::gravity_contributions::Operand = discretization::forms::Operand<Term, Operation>

11.96 src/include/serif/equations/point.hpp File Reference

#include <array>
#include <concepts>
#include <cstddef>
#include <tuple>
#include <type_traits>
#include "serif/discretization/forms/value.hpp"
#include "serif/utils/types/type_list.hpp"
Include dependency graph for point.hpp:



This graph shows which files directly or indirectly include this file:



Classes

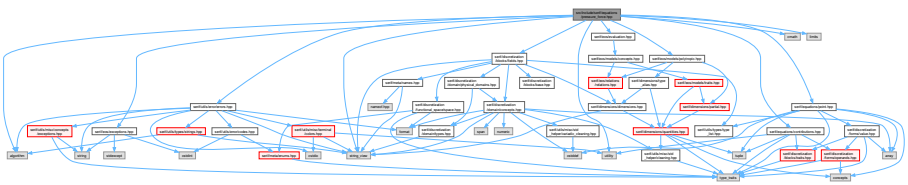
- struct serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >
- struct serif::equations::EquationPoint< OperandsT >
- struct serif::equations::PointLinearization< OperandsT, ResultT >

Namespaces

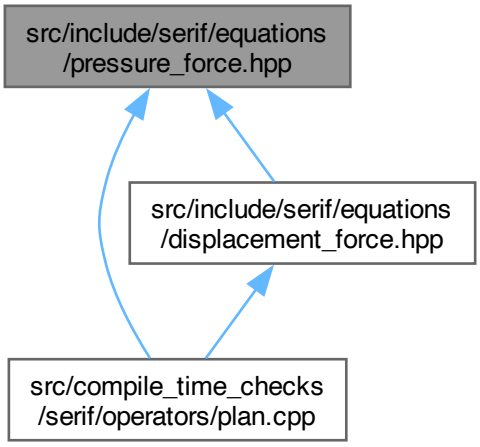
- namespace serif
- namespace serif::equations

11.97 src/include/serif/equations/pressure_force.hpp File Reference

#include <algorithm>
#include <cmath>
#include <limits>
#include <string_view>
#include <utility>
#include "serif/discretization/blocks/fields.hpp"
#include "serif/eos/evaluation.hpp"
#include "serif/eos/exceptions.hpp"
#include "serif/eos/models/polytropic.hpp"
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for pressure_force.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::equations::PressureForce< EOST >

  Pressure's contribution to displacement balance: -integral_star P(h) div(w) dV.

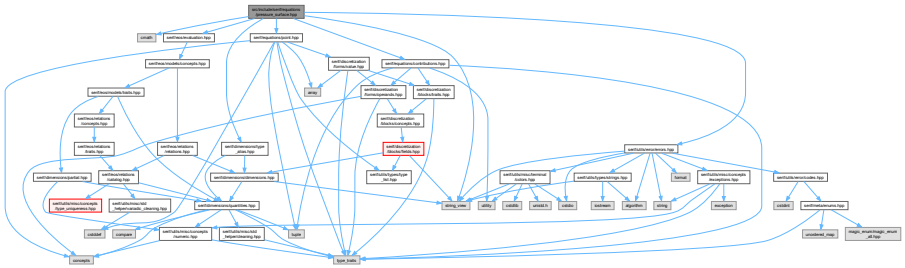
Namespaces

- namespace serif
- namespace serif::equations

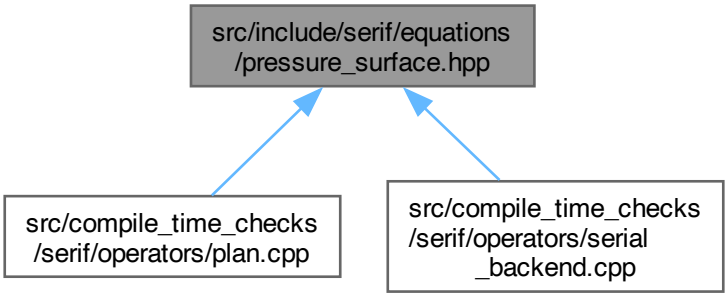
11.98 src/include/serif/equations/pressure_surface.hpp File Reference

#include <cmath>
#include <string_view>
#include "serif/dimensions/type_alias.hpp"
#include "serif/eos/evaluation.hpp"
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"

#include "serif/utils/error/errors.hpp"
Include dependency graph for pressure_surface.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::equations::PressureSurfaceConstraint< EOST >

  Prescribed pressure on the stellar boundary: h - h_EOS(P_surface) = 0.

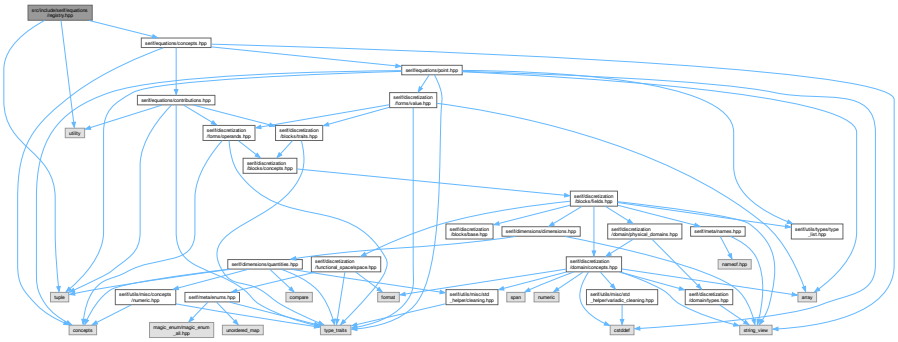
Namespaces

- namespace serif
- namespace serif::equations

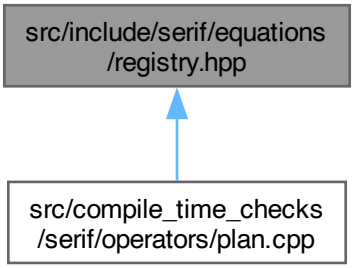
11.99 src/include/serif/equations/registry.hpp File Reference

#include <tuple>
#include <utility>

#include ”serif/equations/concepts.hpp”
Include dependency graph for registry.hpp:



This graph shows which files directly or indirectly include this file:



Classes

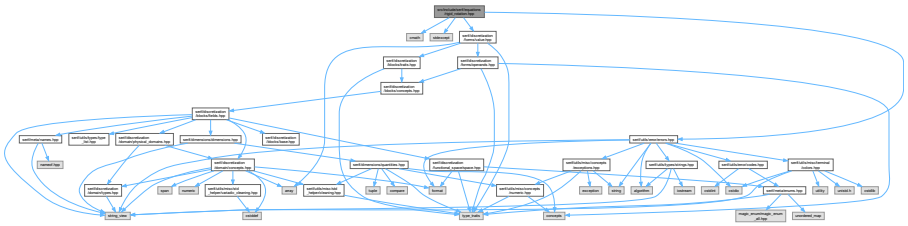
- class serif::equations::EquationRegistry< EquationTs >

Namespaces

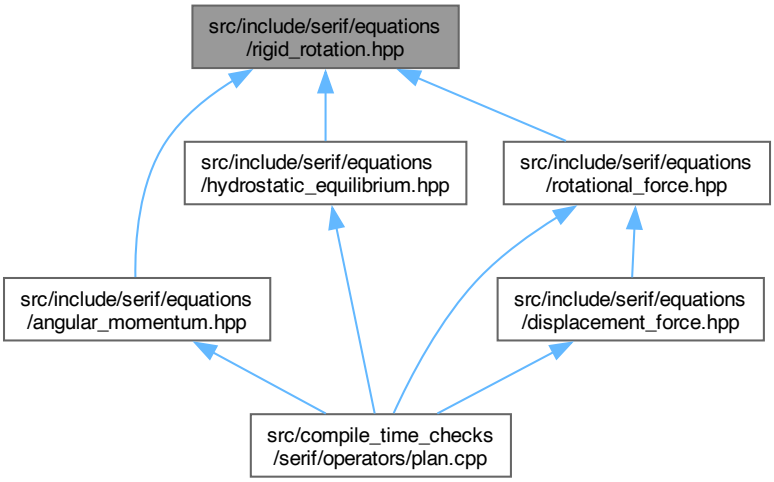
- namespace serif
- namespace serif::equations

11.100 src/include/serif/equations/rigid_rotation.hpp File Reference

#include <cmath>
#include <stdexcept>
#include "serif/discretization/forms/value.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for rigid_rotation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

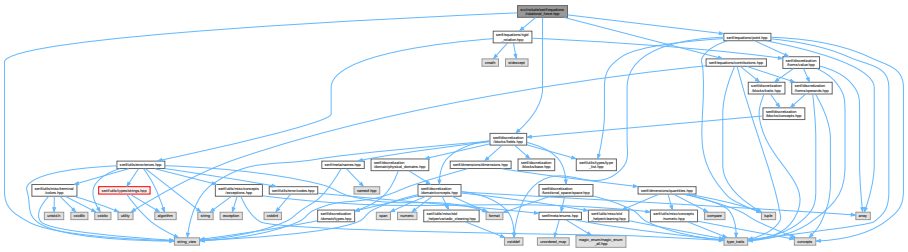
- class serif::equations::RigidRotation

Namespaces

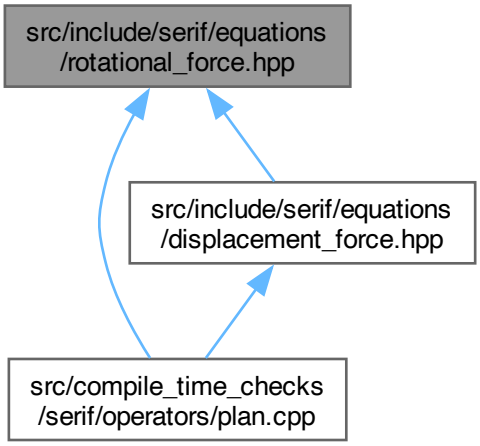
- namespace serif
- namespace serif::equations

11.101 src/include/serif/equations/rotational_force.hpp File Reference

#include <string_view>
#include "serif/discretization/blocks/fields.hpp"
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"
#include "serif/equations/rigid_rotation.hpp"
Include dependency graph for rotational_force.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::equations::RotationalForce

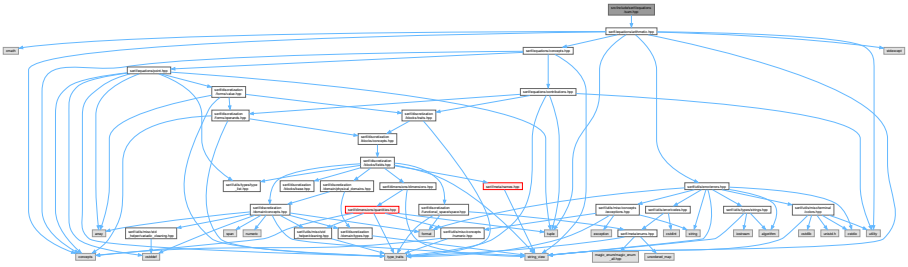
  Rotational contribution to displacement balance: -integral_star rho grad(Psi) . w dV.

Namespaces

- namespace serif
- namespace serif::equations

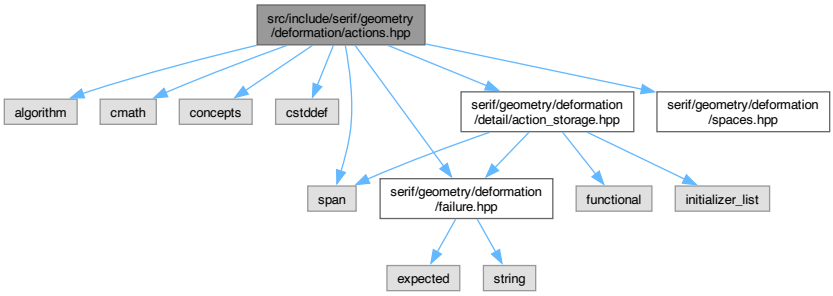
11.102 src/include/serif/equations/sum.hpp File Reference

#include ”serif/equations/arithmetic.hpp”
Include dependency graph for sum.hpp:

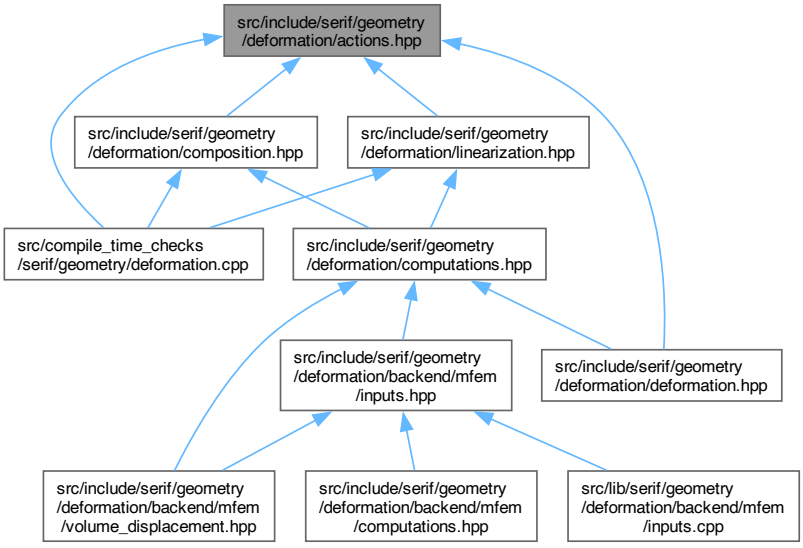


11.103 src/include/serif/geometry/deformation/actions.hpp File Reference

#include <algorithm>
#include <cmath>
#include <concepts>
#include <cstddef>
#include
#include ”serif/geometry/deformation/failure.hpp”
#include ”serif/geometry/deformation/detail/action_storage.hpp”
#include ”serif/geometry/deformation/spaces.hpp”
Include dependency graph for actions.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::EmptyDeformationWorkspace

  Stateless linear actions need no hidden mutable scratch.
- struct serif::geometry::deformation::DeformationWorkspaceType< Map, Linear >

  Select numerical workspace storage without adding a preparation owner.
- struct serif::geometry::deformation::DeformationWorkspaceType< Map, true >

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

Concepts

- concept serif::geometry::deformation::LinearDeformationMap

  Fixed maps supply their forward and work-conjugate transpose once.
- concept serif::geometry::deformation::DifferentiableDeformationMap

  Nonlinear maps explicitly provide every term needed by a Newton force pullback.
- concept serif::geometry::deformation::DeformationMap

Typedefs

- template<DeformationMap Map>
  using serif::geometry::deformation::DeformationWorkspace = typename DeformationWorkspaceType<Map>::Type

Functions

- DeformationStatus serif::geometry::deformation::validate_deformation_values (std::span< const double > values, std::size_t expected_size)

  Reject invalid mathematical arguments before a map touches its output.
- template<DeformationMap Map>
  DeformationWorkspace< Map > serif::geometry::deformation::make_deformation_workspace (const Map &map)

  Allocate action scratch explicitly, once per independent caller.
- template<DeformationMap Map>
  DeformationStatus serif::geometry::deformation::build_displacement (const Map &map, std::span< const double > parameters, std::span< double > displacement, DeformationWorkspace< Map > &workspace)

  Evaluate physical displacement with caller-owned output and scratch. All action outputs must be disjoint from their inputs. Workspace storage is reserved for the map and must not supply public input or output spans.
- template<DeformationMap Map>
  DeformationStatus serif::geometry::deformation::apply_jacobian (const Map &map, std::span< const double > parameters, std::span< const double > direction, std::span< double > displacement_direction, DeformationWorkspace< Map > &workspace)

  Linear maps reuse their primal action; nonlinear maps supply their exact tangent.
- template<DeformationMap Map>
  DeformationStatus serif::geometry::deformation::apply_jacobian_transpose (const Map &map, std::span< const double > parameters, std::span< const double > displacement_dual, std::span< double > parameter_dual, DeformationWorkspace< Map > &workspace)

  Transfer virtual work from the displacement field to its independent shape coefficients.
- template<DeformationMap Map>
  DeformationStatus serif::geometry::deformation::apply_pullback_derivative (const Map &map, std::span< const double > parameters, std::span< const double > direction, std::span< const double > displacement_dual, std::span< double > parameter_dual, DeformationWorkspace< Map > &workspace)

  Differentiate the transpose at fixed dual; zero is justified only for a declared linear map.

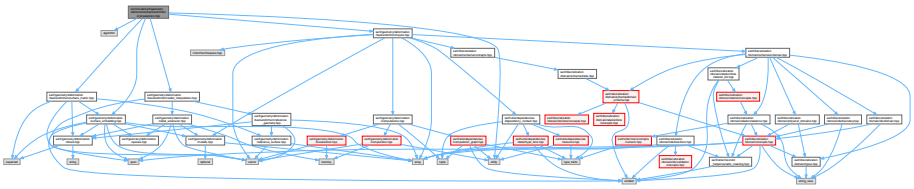
Variables

- template<DeformationMap Map>
  constexpr bool serif::geometry::deformation::is_linear_deformation

  Linearity is a property of the numerical map, independent of the cache lifecycle.

11.104 src/include/serif/geometry/deformation/backend/mfem/computations.hpp File Reference

#include <algorithm>
#include <utility>
#include "serif/geometry/deformation/backend/mfem/inputs.hpp"
#include "serif/geometry/deformation/backend/mfem/radial_interpolation.hpp"
#include "serif/geometry/deformation/backend/mfem/surface_metric.hpp"
#include "serif/geometry/deformation/surface_embedding.hpp"
Include dependency graph for computations.hpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem
- namespace serif::geometry::deformation::backend::mfem::detail

Functions

- template<typename Value>
  std::expected< std::shared_ptr< const Value >, DeformationFailure > serif::geometry::deformation::backend::mfem::det
  (std::expected< Value, DeformationFailure > result)

  Share a successfully compiled immutable table without copying it into each consumer.
- std::expected< RadialInterpolation, DeformationFailure > serif::geometry::deformation::backend::mfem::detail::compile_
  (const ReferenceDiscretization &input)

  Resolve changing mesh labels at preparation time, keeping material masks out of model constructors.
- auto serif::geometry::deformation::backend::mfem::make_surface_metric_computation ()

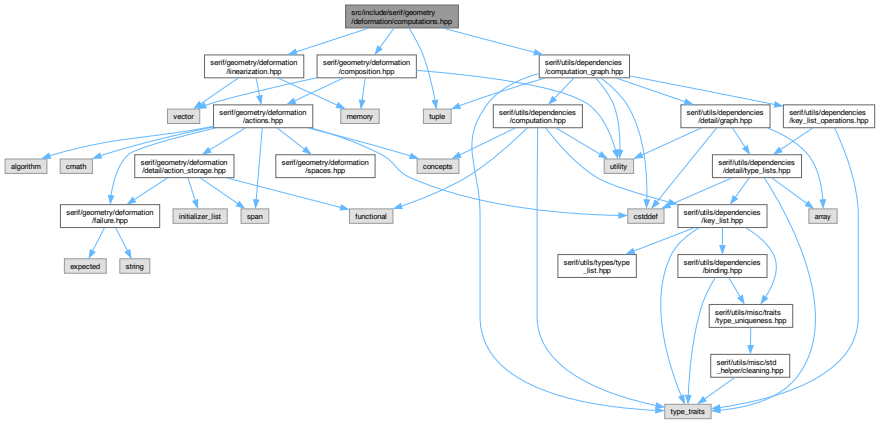
  Optionally supply the reference surface mass diagonal in native shape coordinates.
- template<typename Surface>
  requires requires(const Surface& model, std::span<const std::array<double, 3>> positions) { { build_surface_embedding(model, positions) } -> std::same_as<std::expected<SurfaceEmbedding, DeformationFailure>>; }
  auto serif::geometry::deformation::backend::mfem::make_deformation_computations ()

  Compile fixed reference factors and evaluate shape motion with one dependency engine.
- template<typename Surface>
  auto serif::geometry::deformation::backend::mfem::make_deformation_graph ()

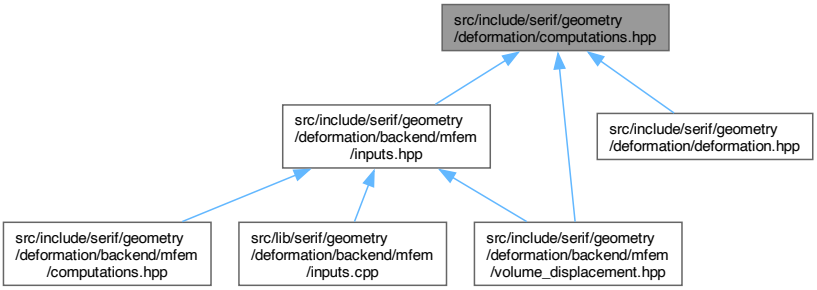
  Provide a standalone graph while leaving the same computations available for equation composition.

11.105 src/include/serif/geometry/deformation/computations.hpp File Reference

#include <tuple>
#include "serif/geometry/deformation/composition.hpp"
#include "serif/geometry/deformation/linearization.hpp"
#include "serif/utils/dependencies/computation_graph.hpp"
Include dependency graph for computations.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::keys::ReferenceDiscretization
- struct serif::geometry::deformation::keys::SurfaceModel
- struct serif::geometry::deformation::keys::InteriorModel
- struct serif::geometry::deformation::keys::VacuumModel
- struct serif::geometry::deformation::keys::ShapeCoefficients
- struct serif::geometry::deformation::keys::ReferenceSurface

- struct serif::geometry::deformation::keys::RadialInterpolation
- struct serif::geometry::deformation::keys::SurfaceMap
- struct serif::geometry::deformation::keys::VolumeExtension
- struct serif::geometry::deformation::keys::DomainMap
- struct serif::geometry::deformation::keys::SurfaceDisplacement
- struct serif::geometry::deformation::keys::VolumeDisplacement
- struct serif::geometry::deformation::keys::Linearization
- struct serif::geometry::deformation::keys::SurfaceMetric

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::keys

Functions

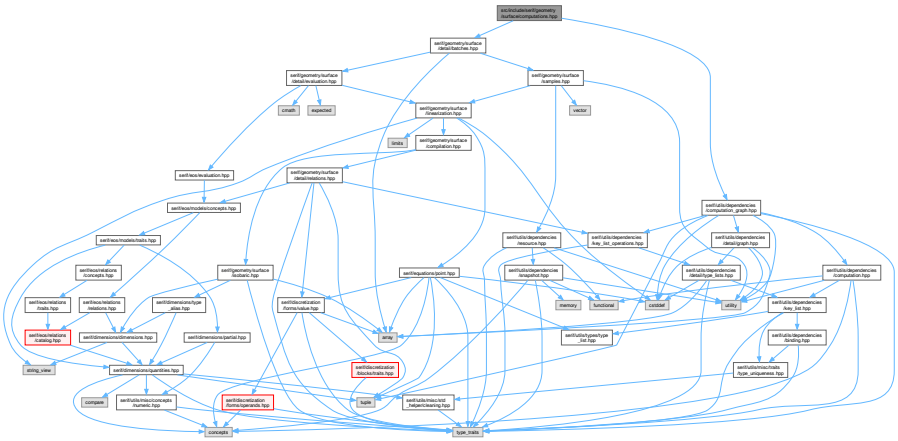
- template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
  requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
  auto serif::geometry::deformation::make_deformation_computations ()

  Declare the numerical composition as ordinary dependency computations.
- template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
  requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
  auto serif::geometry::deformation::make_deformation_graph ()

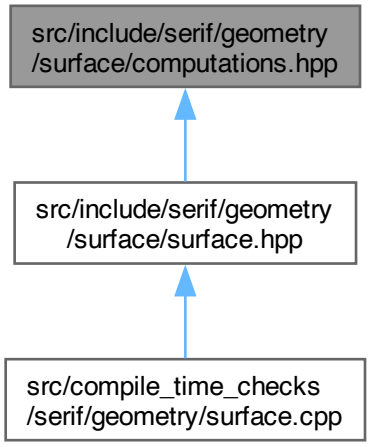
  Compose independently compiled mathematical factors without introducing a prepared-operator facade.

11.106 src/include/serif/geometry/surface/computations.hpp File Reference

#include "serif/geometry/surface/detail/batches.hpp"
#include "serif/utils/dependencies/computation_graph.hpp"
Include dependency graph for computations.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::surface
- namespace serif::geometry::surface::detail

Functions

- template<PressureSurfaceFormulationType Formulation, typename... Terms>
  auto serif::geometry::surface::detail::make_surface_eos_computation (utils::dependencies::KeyList< Terms... >)

  Declare the exact EOS inputs from the formulation, so derivative metadata and cache dependencies cannot diverge.
- template<PressureSurfaceFormulationType Formulation>
  auto serif::geometry::surface::detail::make_surface_linearization_computation ()

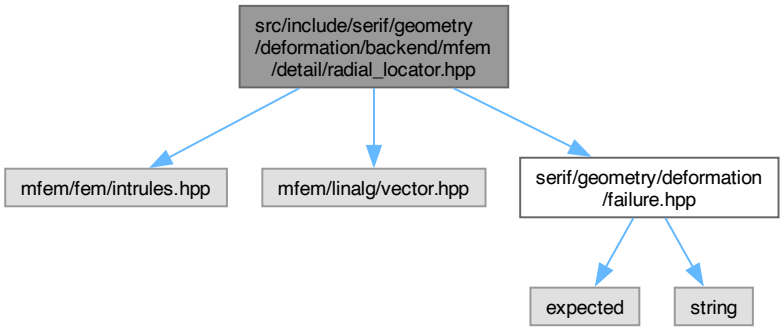
  Keep the inexpensive boundary residual separate from the reusable EOS response and its partials.
- template<PressureSurfaceFormulationType Formulation>
  auto serif::geometry::surface::make_pressure_surface_graph ()

  Describe the isobaric calculation as an ordinary dependency graph, without evaluating the EOS.
- template<typename EOS, typename Bindings, typename Carrier = void>
  requires PressureSurfaceCompilable<EOS, Bindings, Carrier>
  auto serif::geometry::surface::make_pressure_surface_graph ()

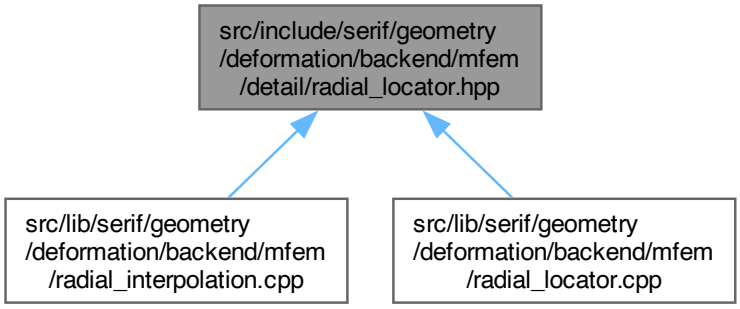
  Infer a unique carrier-resolved formulation directly from an EOS and the available physical state.

11.107 src/include/serif/geometry/deformation/backend/mfem/detail/radial_locator.hpp File Reference

#include ”mfem/fem/intrules.hpp”
#include ”mfem/linalg/vector.hpp”
#include ”serif/geometry/deformation/failure.hpp”
Include dependency graph for radial_locator.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::backend::mfem::detail::LocatedBoundaryPoint
  Keep the best inverse boundary-map result and its physical residual.

Namespaces

- namespace mfem
- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem
- namespace serif::geometry::deformation::backend::mfem::detail

Functions

- double serif::geometry::deformation::backend::mfem::detail::logical_radius (const ::mfem::Vector &position)

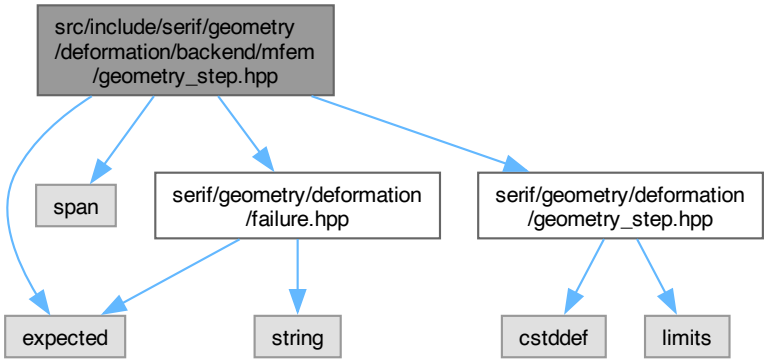
  Preserve the cubical logical radius used by the original radial foliation.
- std::expected< double, DeformationFailure > serif::geometry::deformation::backend::mfem::detail::boundary_logical_ra (::mfem::Mesh &mesh, int attribute)

  Check a complete cubical shell and return its logical radius.
- std::expected< LocatedBoundaryPoint, DeformationFailure > serif::geometry::deformation::backend::mfem::detail::locat (::mfem::Mesh &mesh, int attribute, const ::mfem::Vector &target)

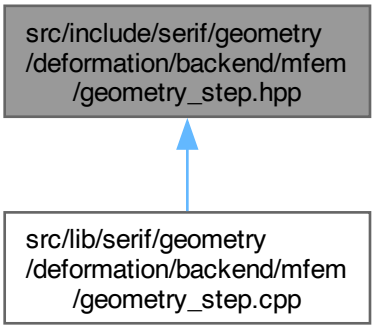
  Locate a projected logical ray on a quadrilateral shell, preserving MeanField tolerances.

11.108 src/include/serif/geometry/deformation/backend/mfem/geometry_step.hpp File Reference

#include <expected>
#include
#include "serif/geometry/deformation/failure.hpp"
#include "serif/geometry/deformation/geometry_step.hpp"
Include dependency graph for geometry_step.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::backend::mfem::GeometrySampleRule

  Borrow one sampling rule; repeat an element to retain the actual union of downstream rules.

Namespaces

- namespace mfem
- namespace serif
- namespace serif::mapping
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

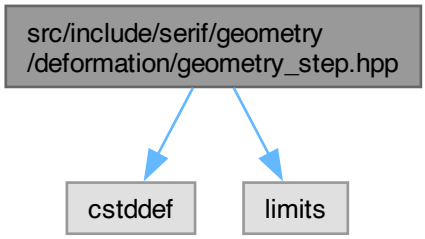
Functions

- std::expected< GeometryStepLimit, DeformationFailure > serif::geometry::deformation::backend::mfem::estimate_geom
  (const mapping::DomainMapper &mapper, const ::mfem::FiniteElementSpace &displacement_↩
  space, const ::mfem::GridFunction &compactification_coordinate, const ::mfem::Vector &accepted_↩
  displacement, const ::mfem::Vector &displacement_direction, std::span< const GeometrySampleRule
  > sample_rules, GeometryStepOptions options={})

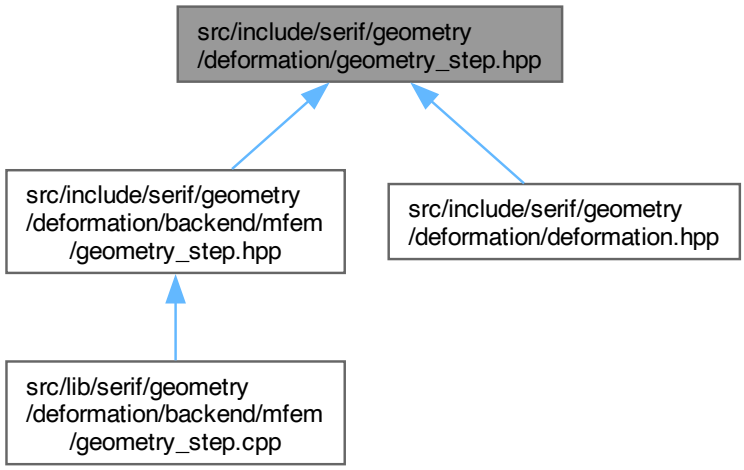
  Find the first sampled mapping boundary along accepted + step direction.

11.109 src/include/serif/geometry/deformation/geometry_step.hpp File Reference

#include <cstddef>
#include <limits>
Include dependency graph for geometry_step.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::GeometryStepOptions

  Numerical controls for keeping a straight volume-displacement step above a Jacobian determinant floor.
- struct serif::geometry::deformation::GeometryStepLimit

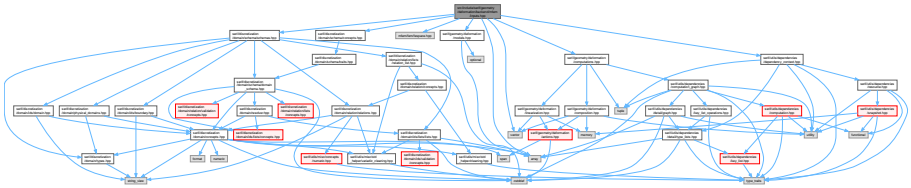
  Geometry-only bound at the supplied sample points, with a location for the first obstruction.

Namespaces

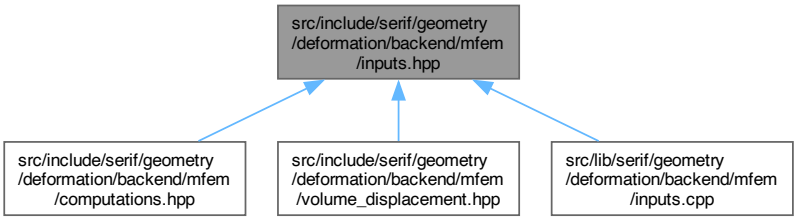
- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

11.110 src/include/serif/geometry/deformation/backend/mfem/inputs.hpp File Reference

#include <array>
#include <vector>
#include "mfem/fem/fespace.hpp"
#include "serif/discretization/domain/schema/schemas.hpp"
#include "serif/discretization/domain/schema/concepts.hpp"
#include "serif/geometry/deformation/models.hpp"
#include "serif/geometry/deformation/computations.hpp"
#include "serif/utils/dependencies/dependency_context.hpp"
Include dependency graph for inputs.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::backend::mfem::ReferenceDiscretization

  Borrow the paired reference meshes; attributes retain their physical domain meaning.
- struct serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot

  Capture coordinate revisions as well as topology and mutable material/boundary labels.

- struct serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot

  Track the coefficient interpretation independently of numerical coefficient values.
- struct serif::geometry::deformation::backend::mfem::ReferenceDiscretizationSnapshot

  One comparable snapshot covers all inputs used to compile fixed deformation data.
- struct serif::geometry::deformation::backend::mfem::ReferenceDiscretizationTracking

  Let the generic dependency engine track MFEM metadata without owning meshes.
- struct serif::geometry::deformation::backend::mfem::DeformationValueTracking

  Model options and coefficient copies describe values independently of wrapper addresses.

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

Functions

- template<discretization::domain::schema::IsSchema Schema = discretization::domain::schema::CoreEnvelopeVacuum↩ DomainSchema>
  requires (Schema::template contains_domain<discretization::domain::StellarDomains>() && Schema::template contains_domain<discretization::domain::VacuumDomain>() && Schema::template contains_boundary<discretization::domain::StellarSurf && Schema::template contains_boundary<discretization::domain::InfinitySurfaceBoundary>())
  ReferenceDiscretization serif::geometry::deformation::backend::mfem::deformation_discretization (const ::mfem::FiniteElementSpace &scalar_space, const ::mfem::FiniteElementSpace &displacement↩ _space, ::mfem::Mesh &logical_mesh)

  Resolve physical domain names through the schema rather than hard-coded MFEM attributes.
- template<typename Surface, typename Interior, typename Vacuum>
  auto serif::geometry::deformation::backend::mfem::make_deformation_inputs (const DomainDeformation< Surface, Interior, Vacuum > &model, const ReferenceDiscretization &discretization, const std::vector< double > &coefficients)

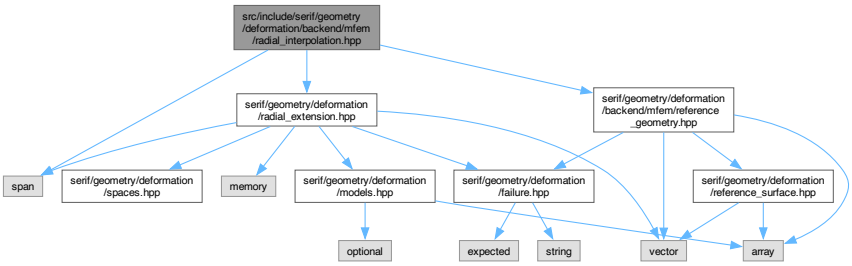
  Bind physical choices and shape coordinates without evaluating deformation.
- template<typename Surface, typename Interior, typename Vacuum>
  void serif::geometry::deformation::backend::mfem::make_deformation_inputs (const DomainDeformation< Surface, Interior, Vacuum > &, const ReferenceDiscretization &, const std::vector< double > &&)=delete

  Reject temporary coefficient vectors whose storage would expire before preparation.

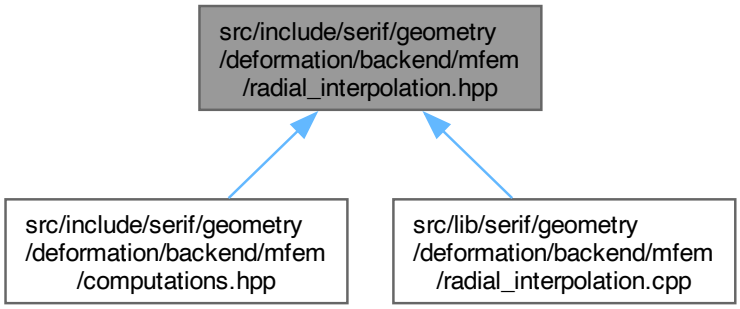
11.111 src/include/serif/geometry/deformation/backend/mfem/radial_↩ interpolation.hpp File Reference

#include
#include "serif/geometry/deformation/radial_extension.hpp"

#include ”serif/geometry/deformation/backend/mfem/reference_geometry.hpp”
Include dependency graph for radial_interpolation.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

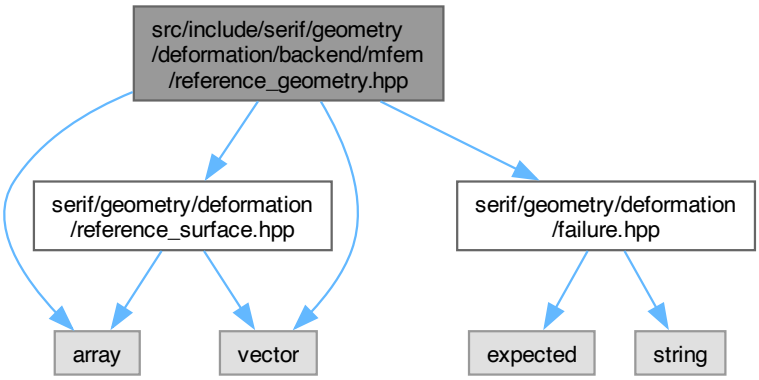
Functions

- std::expected< RadialInterpolation, DeformationFailure > serif::geometry::deformation::backend::mfem::build_radial_i (const ::mfem::FiniteElementSpace &scalar_space, const ::mfem::FiniteElementSpace &vector_space, ::mfem::Mesh &logical_mesh, std::span< const int > stellar_element_support, std::span< const int > vacuum_element_support, int surface_attribute, int infinity_attribute)

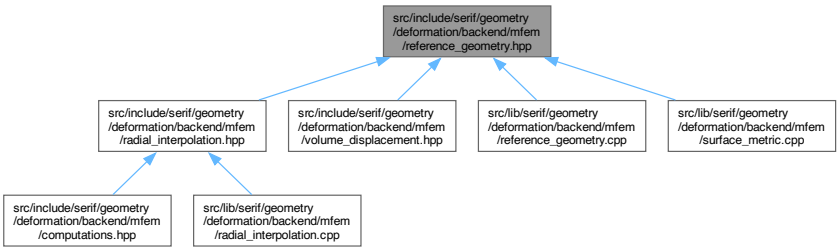
  Compile logical-ray interpolation on paired physical and logical meshes. Supports are zero/one element masks. The logical radius is the maximum absolute Cartesian coordinate, preserving the STROID reference foliation after physical distortion. Only conforming serial nodal spaces and component-block vector ordering are supported.

11.112 src/include/serif/geometry/deformation/backend/mfem/reference_geometry.hpp File Reference

#include <array>
#include <vector>
#include "serif/geometry/deformation/failure.hpp"
#include "serif/geometry/deformation/reference_surface.hpp"
Include dependency graph for reference_geometry.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace mfem
- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

Functions

- DeformationStatus serif::geometry::deformation::backend::mfem::validate_scalar_reference_space (const ::mfem::FiniteElementSpace &scalar_space)

  Reject spaces whose coefficients cannot be interpreted as serial nodal positions.
- DeformationStatus serif::geometry::deformation::backend::mfem::validate_companion_numbering (const ::mfem::FiniteElementSpace &reference_space, const ::mfem::FiniteElementSpace &companion↩ _space)

  Require identical scalar coefficient meanings in two companion spaces. Vector component ordering may differ; element bases and their scalar-node indices must agree.
- std::expected< std::vector< std::array< double, 3 > >, DeformationFailure > serif::geometry::deformation::backend::mfe (const ::mfem::FiniteElementSpace &scalar_space, ::mfem::Mesh &reference_mesh)

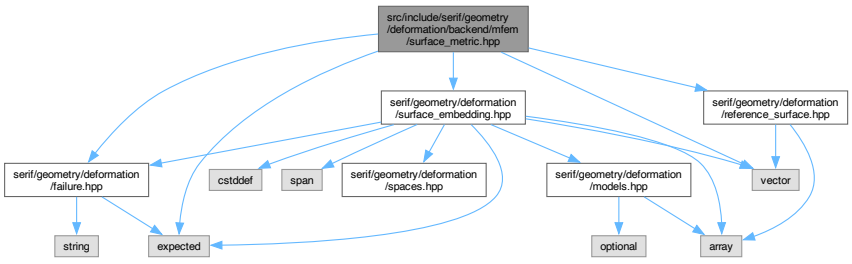
  Sample a paired mesh at the scalar companion nodes, checking shared-node agreement.
- std::expected< std::vector< int >, DeformationFailure > serif::geometry::deformation::backend::mfem::select_surface_s (const ::mfem::FiniteElementSpace &scalar_space, int surface_attribute)

  Select canonical surface nodes without retaining a mesh or finite-element-space pointer.
- std::expected< ReferenceSurface, DeformationFailure > serif::geometry::deformation::backend::mfem::sample_reference (const ::mfem::FiniteElementSpace &scalar_space, int surface_attribute)

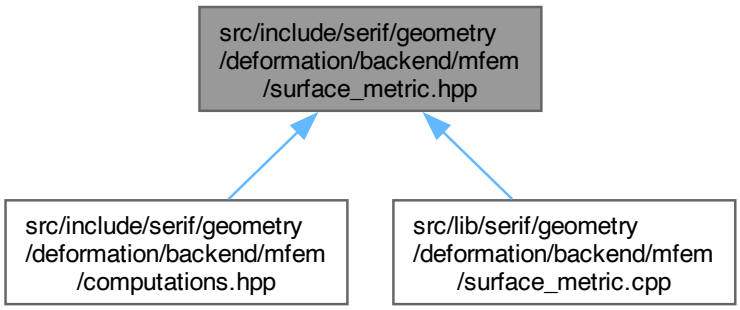
  Sample physical reference positions in ascending scalar-DOF order.

11.113 src/include/serif/geometry/deformation/backend/mfem/surface_↩ metric.hpp File Reference

#include <expected>
#include <vector>
#include "serif/geometry/deformation/failure.hpp"
#include "serif/geometry/deformation/reference_surface.hpp"
#include "serif/geometry/deformation/surface_embedding.hpp"
Include dependency graph for surface_metric.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace mfem
- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

Functions

- std::expected< std::vector< double >, DeformationFailure > serif::geometry::deformation::backend::mfem::surface_para (const ::mfem::FiniteElementSpace &scalar_space, int surface_attribute, const ReferenceSurface &reference_surface, const SurfaceEmbedding &embedding)

  Measure each shape parameter with the reference boundary mass norm of its Cartesian finite-element trace.

11.114 src/include/serif/geometry/deformation/backend/mfem/volume_displacement.hpp File Reference

#include <algorithm>
#include <optional>
#include <tuple>
#include "serif/geometry/deformation/computations.hpp"
#include "serif/geometry/deformation/backend/mfem/inputs.hpp"
#include "serif/geometry/deformation/backend/mfem/reference_geometry.hpp"

#include ”serif/operators/backend/mfem/volume_inputs.hpp”
Include dependency graph for volume_displacement.hpp:



Classes

- struct serif::geometry::deformation::backend::mfem::volume_displacement_keys::CheckedBinding
- struct serif::geometry::deformation::backend::mfem::volume_displacement_keys::Coefficients

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem
- namespace serif::geometry::deformation::backend::mfem::volume_displacement_keys

Functions

- template<discretization::domain::schema::IsSchema Schema>
  auto serif::geometry::deformation::backend::mfem::make_volume_displacement_computations (const discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, Schema > &binding)

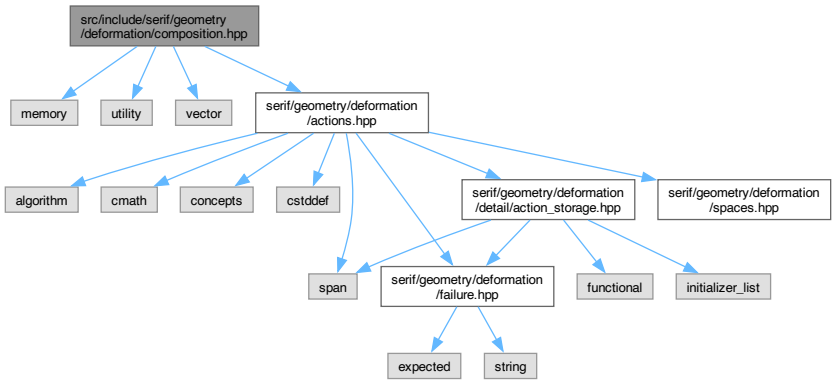
  Connect a full component-block deformation vector to an equation's displacement binding.
- template<discretization::domain::schema::IsSchema Schema>
  void serif::geometry::deformation::backend::mfem::make_volume_displacement_computations (const discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, Schema > &&)=delete

  A graph cannot retain a displacement binding supplied as a temporary.

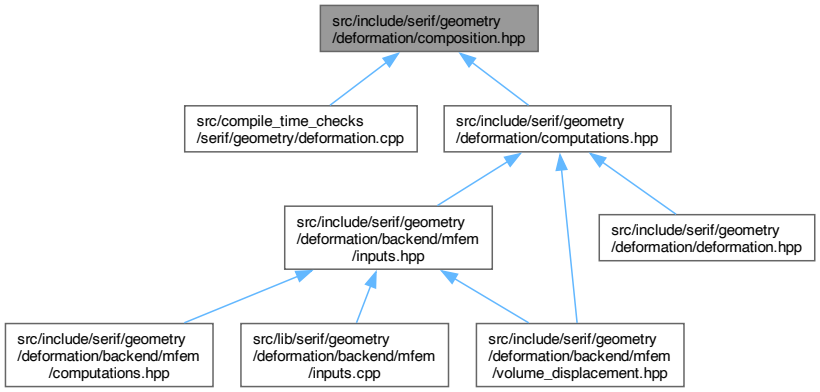
11.115 src/include/serif/geometry/deformation/composition.hpp File Reference

#include <memory>
#include <utility>
#include <vector>

#include ”serif/geometry/deformation/actions.hpp”
Include dependency graph for composition.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >
  Compose shape-to-surface and surface-to-volume maps with the complete derivative chain rule.
- struct serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

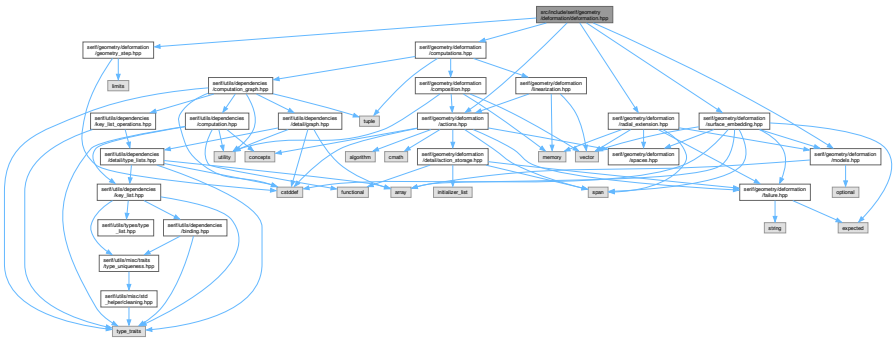
Functions

- template<DeformationMap SurfaceMap, DeformationMap ExtensionMap>
  requires std::same_as<typename SurfaceMap::OutputSpace, typename ExtensionMap::InputSpace>
  std::expected< ComposedDeformation< SurfaceMap, ExtensionMap >, DeformationFailure > serif::geometry::deformation::compose_deformation (std::shared_ptr< const SurfaceMap > surface, std::shared_ptr< const ExtensionMap > extension)

  Validate factor sizes before constructing an owning mathematical composition.

11.116 src/include/serif/geometry/deformation/deformation.hpp File Reference

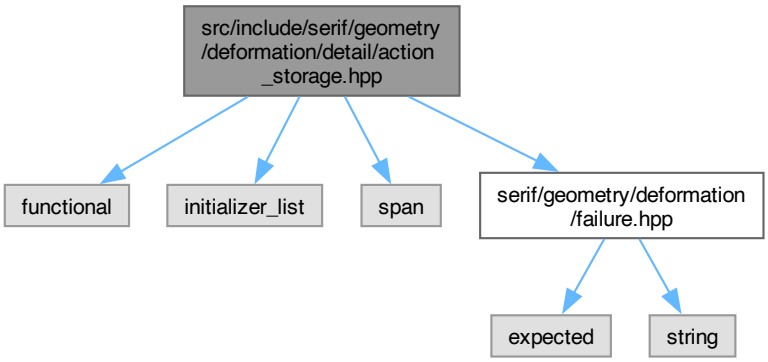
#include "serif/geometry/deformation/actions.hpp"
#include "serif/geometry/deformation/computations.hpp"
#include "serif/geometry/deformation/geometry_step.hpp"
#include "serif/geometry/deformation/models.hpp"
#include "serif/geometry/deformation/radial_extension.hpp"
#include "serif/geometry/deformation/surface_embedding.hpp"
Include dependency graph for deformation.hpp:



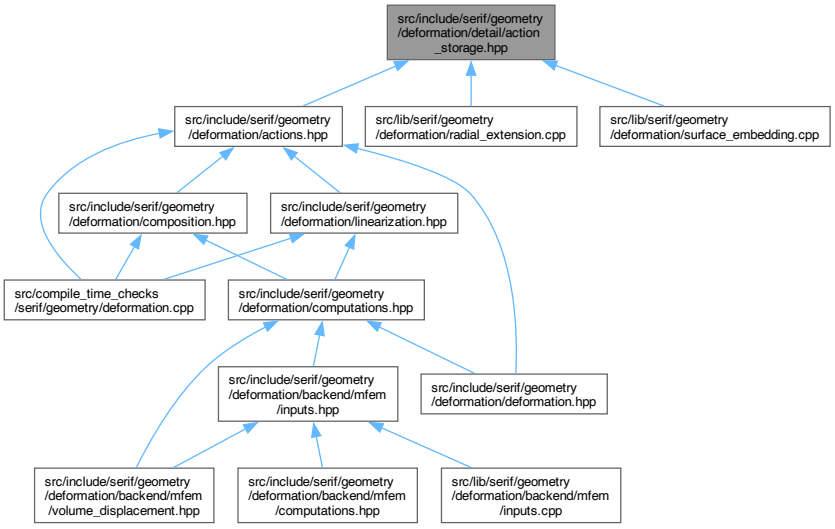
11.117 src/include/serif/geometry/deformation/detail/action_storage.hpp File Reference

#include <functional>
#include <initializer_list>
#include
#include "serif/geometry/deformation/failure.hpp"

Include dependency graph for action_storage.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::detail

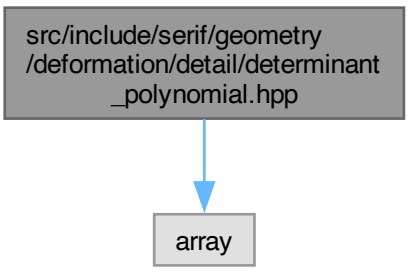
Functions

- DeformationStatus serif::geometry::deformation::detail::validate_disjoint_output (std::span< double > output, std::initializer_list< std::span< const double > > inputs)

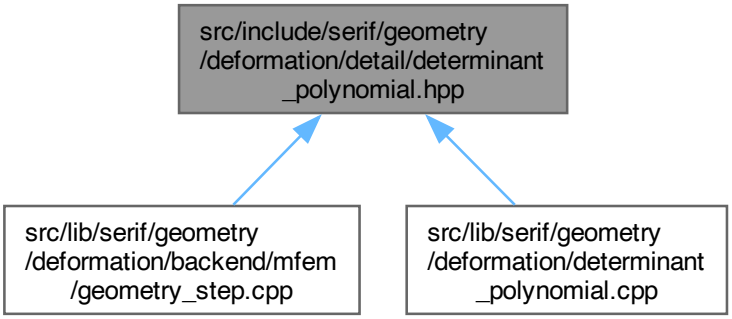
  Preserve every action input until the last use, including nonlinear transpose derivatives. The list contains only the fixed number of mathematical arguments to an action; checking address intervals requires no allocation and never scans numerical coefficients.

11.118 src/include/serif/geometry/deformation/detail/determinant_↩ polynomial.hpp File Reference

#include <array>
Include dependency graph for determinant_polynomial.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::detail

Typedefs

- using serif::geometry::deformation::detail::DeterminantPolynomial = std::array<double, 4>
- using serif::geometry::deformation::detail::JacobianEntries = std::array<std::array<double, 3>, 3>

Functions

- DeterminantPolynomial serif::geometry::deformation::detail::determinant_polynomial (const JacobianEntries &base, const JacobianEntries &direction, int dimension, double maximum_step, double determinant↩ _floor) noexcept

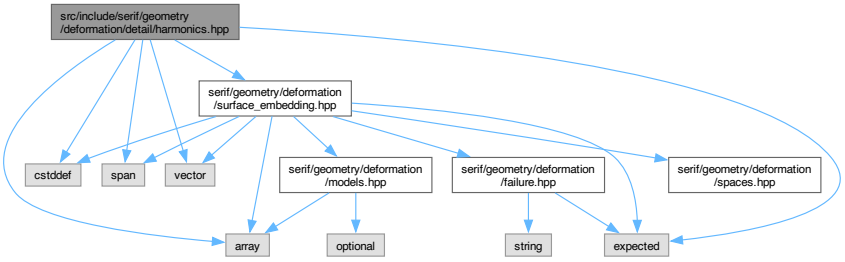
  Expand det(base + t maximum_step direction) minus the floor for dimensions one through three.
- double serif::geometry::deformation::detail::evaluate_polynomial (const DeterminantPolynomial &coefficients, double parameter) noexcept

  Evaluate the cubic with fused operations, retaining small values near repeated roots.
- double serif::geometry::deformation::detail::first_boundary_parameter (const DeterminantPolynomial &coefficients) noexcept

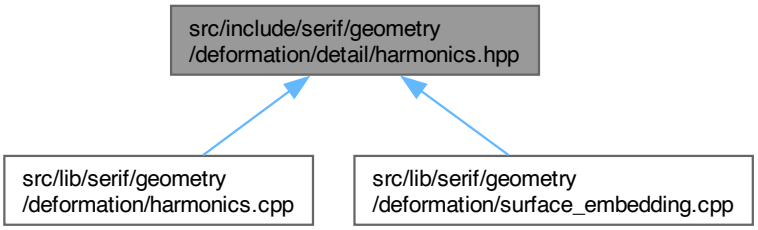
  Find the first loss of strict positivity in (0, 1], including a tangent zero.

11.119 src/include/serif/geometry/deformation/detail/harmonics.hpp File Reference

#include <array>
#include <cstddef>
#include <expected>
#include
#include <vector>
#include ”serif/geometry/deformation/surface_embedding.hpp”
Include dependency graph for harmonics.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::detail

Functions

- std::expected< std::vector< SphericalHarmonicMode >, DeformationFailure > serif::geometry::deformation::detail::mak (const SphericalHarmonicSurface &model, std::size_t node_count)

  Validate harmonic cutoffs before allocating a basis or enumerating modes.
- std::expected< std::vector< double >, DeformationFailure > serif::geometry::deformation::detail::sample_harmonic_ba (const SphericalHarmonicSurface &model, std::span< const SphericalHarmonicMode > modes, std↩ ::span< const std::array< double, 3 > > directions)

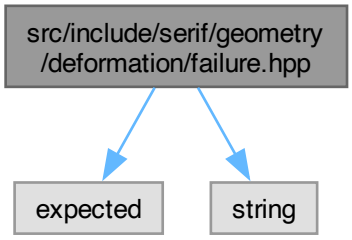
  Use normalized Legendre recurrence to avoid factorial growth while preserving the Condon–Shortley phase.
- DeformationStatus serif::geometry::deformation::detail::check_independent_harmonics (std::span< const double > basis, std::size_t node_count, std::size_t mode_count)

  Check sampled rank on a private copy, leaving physical coefficients and basis normalization unchanged.

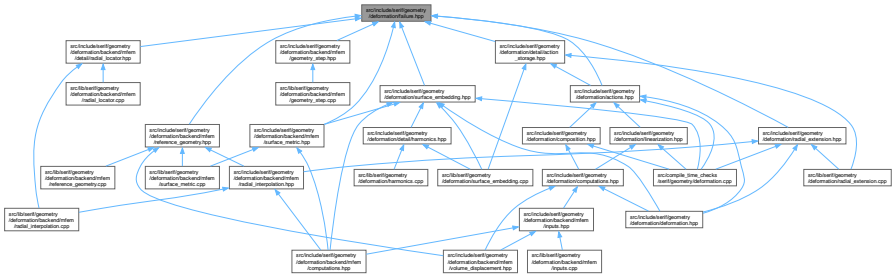
11.120 src/include/serif/geometry/deformation/failure.hpp File Reference

#include <expected>
#include <string>

Include dependency graph for failure.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::DeformationFailure

  Describe a rejected geometry calculation; the dependency cache publishes results only after success.

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

Typedefs

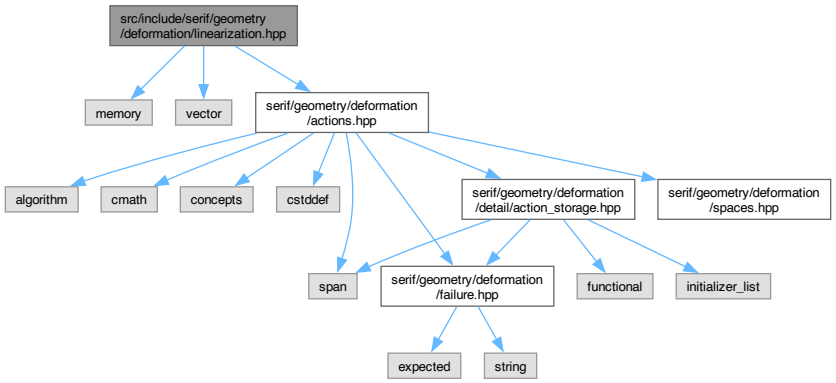
- using serif::geometry::deformation::DeformationStatus = std::expected<void, DeformationFailure>

Enumerations

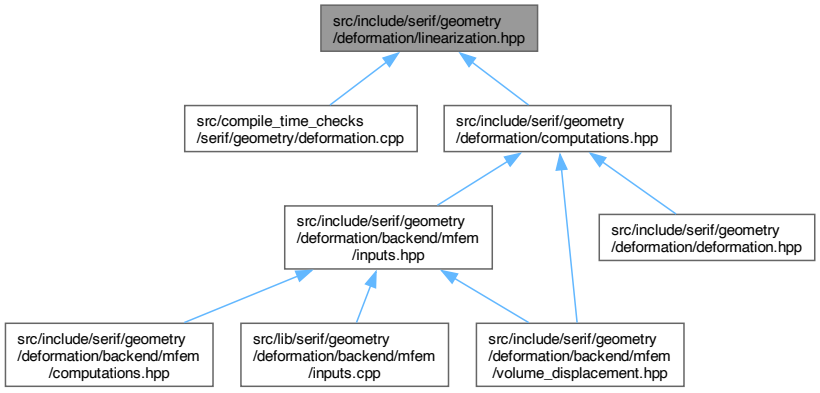
- enum class serif::geometry::deformation::DeformationFailureReason { serif::geometry::deformation::invalid_configuration , serif::geometry::deformation::incompatible_discretization , serif::geometry::deformation::invalid_size , serif::geometry::deformation::non_finite , serif::geometry::deformation::unresolved_surface , serif::geometry::deformation::invalid_geometry }

11.121 src/include/serif/geometry/deformation/linearization.hpp File Reference

#include <memory>
#include <vector>
#include "serif/geometry/deformation/actions.hpp"
Include dependency graph for linearization.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::DeformationLinearization< Map >

  Freeze the shape coordinates associated with a derivative, independently of mutable caller inputs.

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

Functions

- template<DeformationMap Map>
  std::expected< std::vector< double >, DeformationFailure > serif::geometry::deformation::evaluate_displacement (const Map &map, std::span< const double > parameters)

  Allocate a result for graph publication while leaving repeated low-level actions allocation-free.
- template<DeformationMap Map>
  DeformationStatus serif::geometry::deformation::apply_jacobian (const DeformationLinearization< Map > &state, std::span< const double > direction, std::span< double > output, DeformationWorkspace< Map > &workspace)

  Evaluate a tangent about the cache's accepted shape, not the caller's next trial state.
- template<DeformationMap Map>
  DeformationStatus serif::geometry::deformation::apply_jacobian_transpose (const DeformationLinearization< Map > &state, std::span< const double > dual, std::span< double > output, DeformationWorkspace< Map > &workspace)

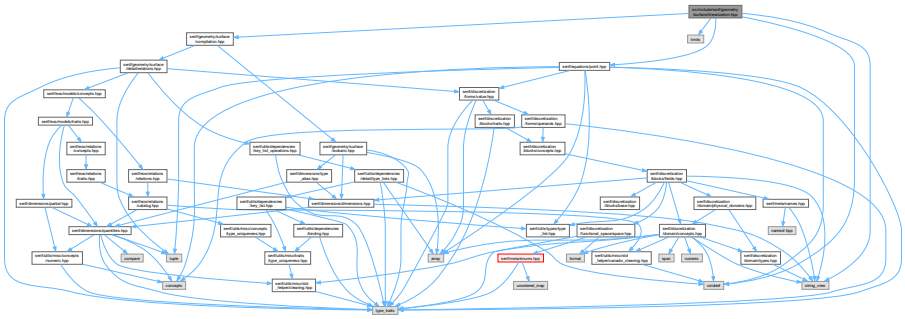
  Pull displacement forces back to the accepted independent shape coordinates.
- template<DeformationMap Map>
  DeformationStatus serif::geometry::deformation::apply_pullback_derivative (const DeformationLinearization< Map > &state, std::span< const double > direction, std::span< const double > dual, std::span< double > output, DeformationWorkspace< Map > &workspace)

  Keep the geometric contribution to the Newton force derivative available for nonlinear models.

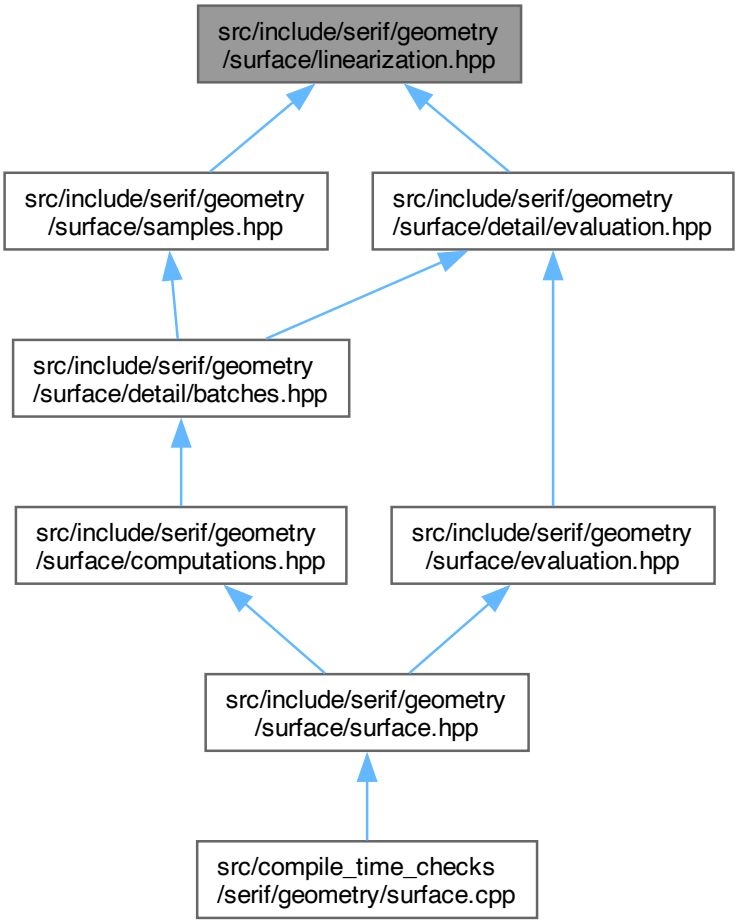
11.122 src/include/serif/geometry/surface/linearization.hpp File Reference

```
#include <cstddef>
#include <limits>
#include <string_view>
#include "serif/equations/point.hpp"
```

#include ”serif/geometry/surface/compilation.hpp”
Include dependency graph for linearization.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::surface::SurfaceFailure
  A rejected thermodynamic candidate, with a sample index when evaluating a boundary batch.
- struct serif::geometry::surface::SurfaceEOSResponse< Formulation >
  An EOS value and all of its local derivatives, evaluated together at one state.

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::surface

Typedefs

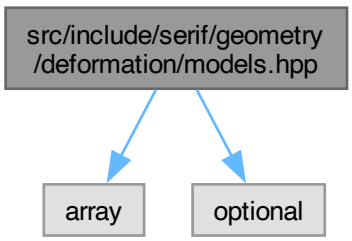
- template<PressureSurfaceFormulationType Formulation>
  using serif::geometry::surface::SurfacePoint = equations::EquationPoint<typename Formulation::↩ Operands>
- template<PressureSurfaceFormulationType Formulation>
  using serif::geometry::surface::SurfaceLinearization = equations::PointLinearization<typename Formulation::Operands>

Enumerations

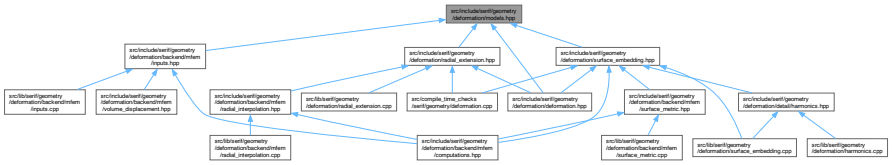
- enum class serif::geometry::surface::SurfaceFailureReason {
  serif::geometry::surface::non_finite_state , serif::geometry::surface::non_finite_eos_response , serif::geometry::surface::non_finite_linearization , serif::geometry::surface::inconsistent_sample_count ,
  serif::geometry::surface::empty_surface }

11.123 src/include/serif/geometry/deformation/models.hpp File Reference

#include <array>
#include <optional>
Include dependency graph for models.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::NodalRadialSurface

  Give each surface node an independent radial displacement, measured in physical length.
- struct serif::geometry::deformation::SphericalHarmonicSurface

  Represent radial displacement lengths with normalized real spherical harmonics on the reference surface.
- struct serif::geometry::deformation::PowerLawRadialExtension

  Extend surface motion toward a fixed stellar center using a power of logical reference radius.
- struct serif::geometry::deformation::FixedInfinityRadialExtension

  Reduce surface motion linearly to zero at reference infinity in logical reference radius.
- struct serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >

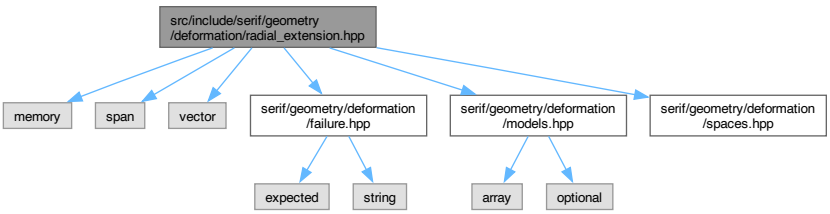
  State how the stellar surface moves and how that motion extends into the two surrounding regions.

Namespaces

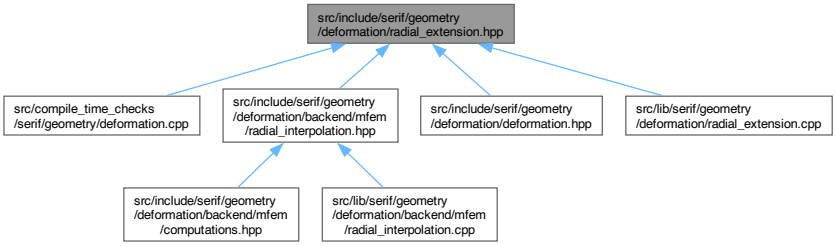
- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

11.124 src/include/serif/geometry/deformation/radial_extension.hpp File Reference

#include <memory>
#include
#include <vector>
#include "serif/geometry/deformation/failure.hpp"
#include "serif/geometry/deformation/models.hpp"
#include "serif/geometry/deformation/spaces.hpp"
Include dependency graph for radial_extension.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::RadialInterpolation

  Own the surface interpolation shared by the stellar and vacuum extensions.
- class serif::geometry::deformation::RadialExtension

  Extend one interleaved surface vector into the MFEM component-block volume layout. Stellar ownership wins at shared nodes. Both actions use the same scalar interpolation, so their pairing is Euclidean virtual work without an additional surface-area weight.

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

Functions

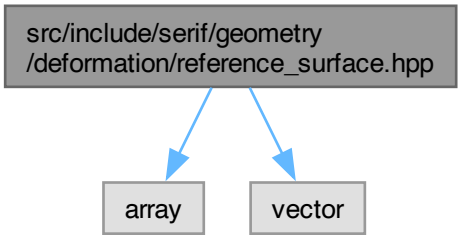
- std::expected< RadialExtension, DeformationFailure > serif::geometry::deformation::make_radial_extension (std::shared_ptr< const RadialInterpolation > interpolation, PowerLawRadialExtension interior={}, FixedInfinityRadialExtension vacuum={})

  Compile regional weights while sharing the immutable surface interpolation. Callers supplying a shared pointer must treat its data as immutable after compilation; the dependency graph publishes a const owning value for precisely this reason.

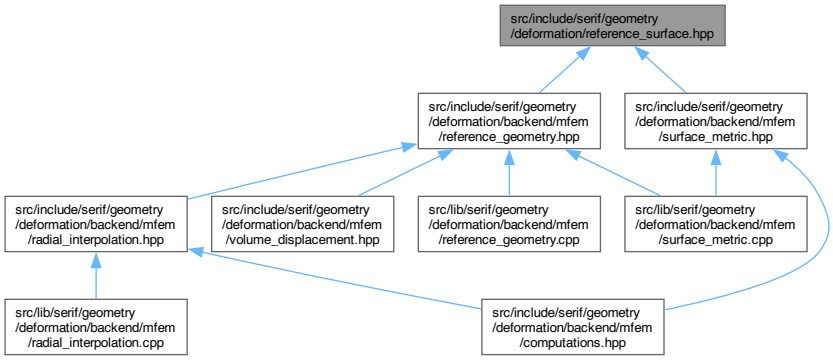
11.125 src/include/serif/geometry/deformation/reference_surface.hpp File Reference

#include <array>
#include <vector>

Include dependency graph for reference_surface.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::ReferenceSurface

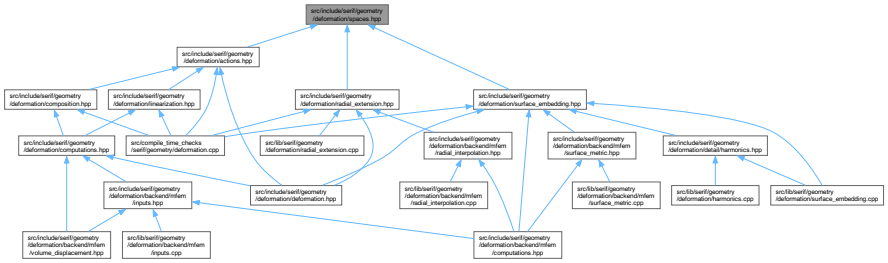
  Own fixed surface positions and their canonical ascending scalar-node indices.

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

11.126 src/include/serif/geometry/deformation/spaces.hpp File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::spaces::ShapeParameters

  Independent coordinates belonging to the selected surface representation.
- struct serif::geometry::deformation::spaces::SurfaceDisplacement

  Cartesian surface traces with components interleaved at each canonical node.
- struct serif::geometry::deformation::spaces::VolumeDisplacement

  Displacement coefficients in the declared volume discretization.

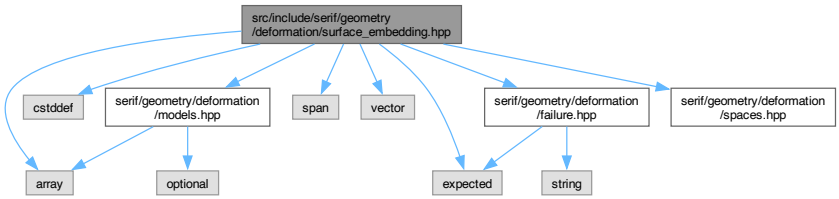
Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::spaces

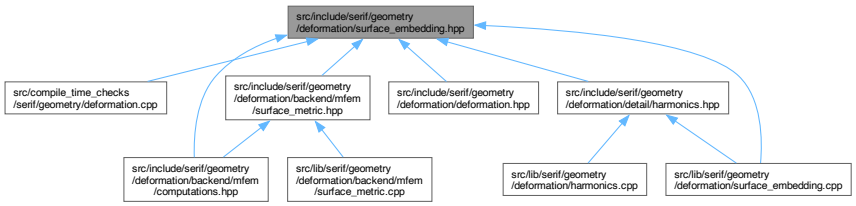
11.127 src/include/serif/geometry/deformation/surface_embedding.hpp File Reference

#include <array>
#include <cstddef>
#include <expected>
#include
#include <vector>
#include "serif/geometry/deformation/failure.hpp"
#include "serif/geometry/deformation/models.hpp"

#include ”serif/geometry/deformation/spaces.hpp”
Include dependency graph for surface_embedding.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::deformation::SphericalHarmonicMode

  Identify a real harmonic, with negative orders using sine and positive orders using cosine.
- class serif::geometry::deformation::SurfaceEmbedding

  Own a linear conversion from radial displacement lengths to Cartesian surface displacements.

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

Functions

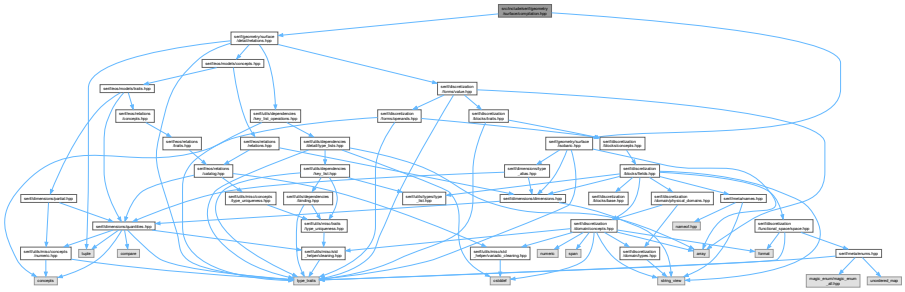
- std::expected< SurfaceEmbedding, DeformationFailure > serif::geometry::deformation::build_surface_embedding (const NodalRadialSurface &model, std::span< const std::array< double, 3 > > positions)

  Compile fixed radial directions from ordered physical reference positions without retaining backend objects.
- std::expected< SurfaceEmbedding, DeformationFailure > serif::geometry::deformation::build_surface_embedding (const SphericalHarmonicSurface &model, std::span< const std::array< double, 3 > > positions)

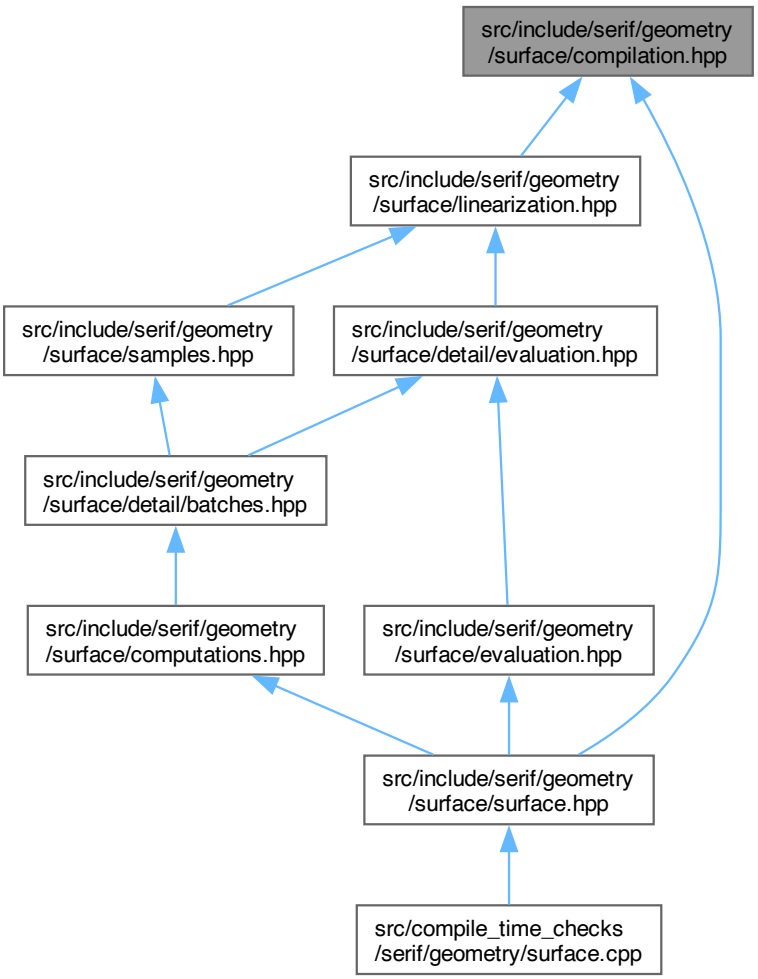
  Sample physically normalized real harmonics and reject unresolved modes without renormalizing or dropping coordinates.

11.128 src/include/serif/geometry/surface/compilation.hpp File Reference

#include ”serif/geometry/surface/detail/relations.hpp”
#include ”serif/geometry/surface/isobaric.hpp”
Include dependency graph for compilation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveC
  Common physical metadata; numerical values and preparation belong to the dependency context and cache.
- struct serif::geometry::surface::PressureSurfaceFormulation< EOS, Bindings, Carrier >
  The boundary residual is carrier minus the EOS value at prescribed pressure; no carrier quantity is privileged.
- struct serif::geometry::surface::DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation >
  Enforce P_EOS(state) - P_surface = 0 without inventing an inverse EOS relation.

Namespaces

- namespace serif

- namespace serif::geometry
- namespace serif::geometry::surface
- namespace serif::geometry::surface::detail

Concepts

- concept serif::geometry::surface::PressureSurfaceCompilable

  Whether the available state selects exactly one pressure-to-carrier EOS relation.
- concept serif::geometry::surface::DirectPressureSurfaceCompilable

  A direct pressure residual requires an explicitly chosen pressure relation and an input quantity whose equation row it replaces.
- concept serif::geometry::surface::PressureSurfaceFormulationType

Typedefs

- template<typename EOS, typename Bindings, typename Carrier = void>
  using serif::geometry::surface::pressure_relation_t = std::tuple_element_t<0, typename detail::MatchingPressureRelat ::remove_cvref_t<EOS>, std::remove_cvref_t<Bindings>, Carrier, typename std::remove_cvref_↩ t<EOS>::Relations>::Matches>

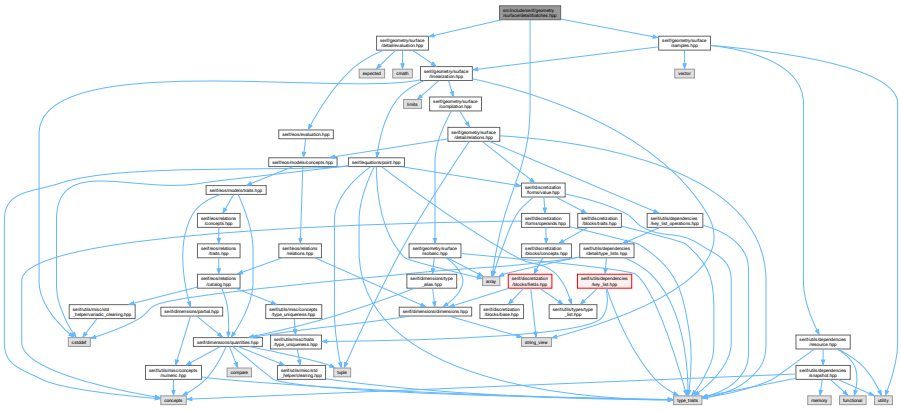
Variables

- template<typename Candidate>
  constexpr bool serif::geometry::surface::detail::is_pressure_surface_formulation = false
- template<typename EOS, typename Bindings, typename Carrier>
  constexpr bool serif::geometry::surface::detail::is_pressure_surface_formulation< PressureSurfaceFormulation< EOS, = true
- template<typename EOS, typename Bindings, typename RowQuantity, typename Relation>
  constexpr bool serif::geometry::surface::detail::is_pressure_surface_formulation< DirectPressureSurfaceFormulation< = true

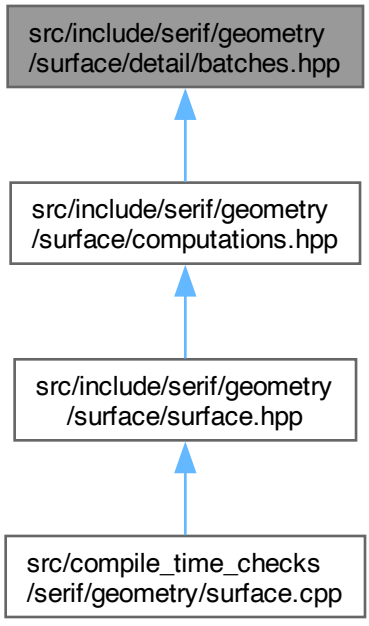
11.129 src/include/serif/geometry/surface/detail/batches.hpp File Reference

#include <array>
#include "serif/geometry/surface/detail/evaluation.hpp"

#include ”serif/geometry/surface/samples.hpp”
Include dependency graph for batches.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::surface
- namespace serif::geometry::surface::detail

Functions

- template<typename... Terms>
  std::expected< std::size_t, SurfaceFailure > serif::geometry::surface::detail::surface_sample_count (const BoundaryValues< Terms > &... columns)

  All EOS columns must describe the same ordered, nonempty set of boundary points.
- template<PressureSurfaceFormulationType Formulation, typename... Terms>
  SurfacePoint< Formulation > serif::geometry::surface::detail::surface_point (const std::size_t sample, const BoundaryValues< Terms > &... columns)

  Assemble the conventional equation point from typed columns; no mesh assumptions enter the physical calculation.
- template<PressureSurfaceFormulationType Formulation, typename... Terms>
  std::expected< std::vector< SurfaceEOSResponse< Formulation > >, SurfaceFailure > serif::geometry::surface::detail::e (const typename Formulation::EquationOfState &equation_of_state, const dimensions::PressureValue pressure, const BoundaryValues< Terms > &... columns)

  Evaluate a whole boundary batch in one graph node, preserving the first failed sample and publishing no partial batch.
- template<PressureSurfaceFormulationType Formulation>
  std::expected< std::vector< SurfaceLinearization< Formulation > >, SurfaceFailure > serif::geometry::surface::detail::li (const std::vector< SurfaceEOSResponse< Formulation > > &responses, const BoundaryValues< typename Formulation::CarrierTerm > &carrier)

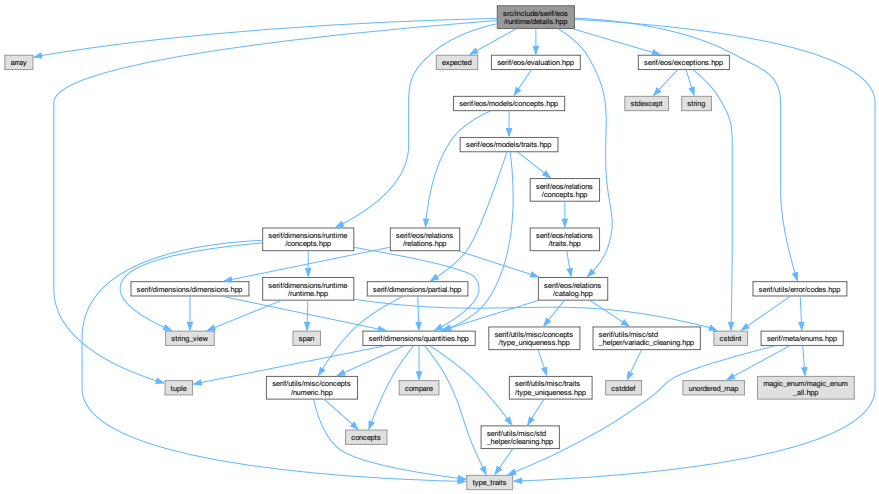
  Broadcast only a genuinely pressure-only relation; otherwise every carrier sample needs its own EOS response.
- template<PressureSurfaceFormulationType Formulation>
  std::expected< std::vector< SurfaceLinearization< Formulation > >, SurfaceFailure > serif::geometry::surface::detail::li (const IsobaricSurface &condition, const std::vector< SurfaceEOSResponse< Formulation > > &responses)

  Changing the prescribed pressure of a direct-pressure residual reuses all EOS values and partials.

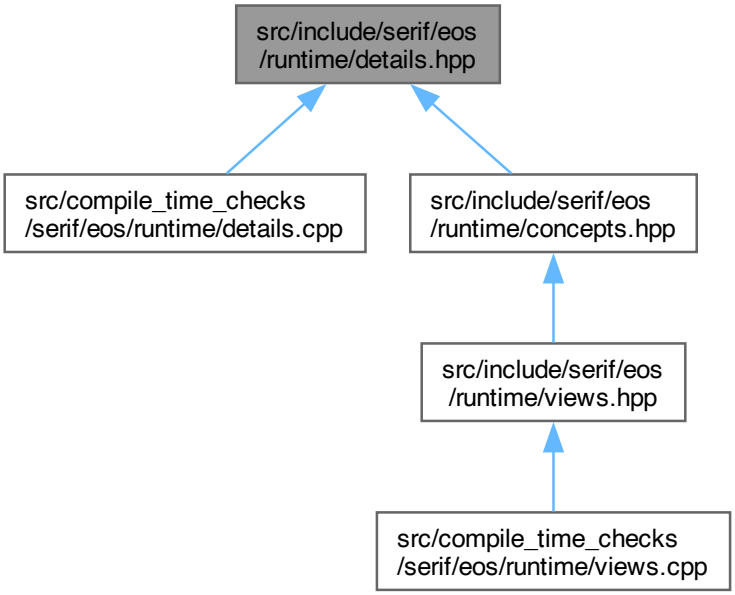
11.130 src/include/serif/eos/runtime/details.hpp File Reference

#include <array>
#include <type_traits>
#include <tuple>
#include <expected>
#include "serif/dimensions/runtime/concepts.hpp"
#include "serif/eos/exceptions.hpp"
#include "serif/eos/evaluation.hpp"
#include "serif/eos/relations/catalog.hpp"
#include "serif/utils/error/codes.hpp"

Include dependency graph for details.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::eos::runtime::HasRuntimeQuantityIdentifiers< RelationT >

- struct serif::eos::runtime::HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > >
- struct serif::eos::runtime::RuntimeRelationQuantities< relations::Relation< OutputT, InputTs... > >
- struct serif::eos::runtime::RuntimeRelationsAreSupported< QuantitiesAreIdentified, RelationTs >
- struct serif::eos::runtime::RuntimeRelationsAreSupported< true, RelationTs... >
- struct serif::eos::runtime::RuntimeCatalogIsSupported< CatalogT >
- struct serif::eos::runtime::RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > >
- struct serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >
- struct serif::eos::runtime::RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > >
- struct serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > >

Namespaces

- namespace serif
- namespace serif::eos
- namespace serif::eos::runtime

Typedefs

- template<typename... RelationTs>
  using serif::eos::runtime::RuntimeCatalogQuantityTuple = decltype(std::tuple_cat(std::declval<typename RuntimeRelationQuantities<RelationTs>::Type>()...))

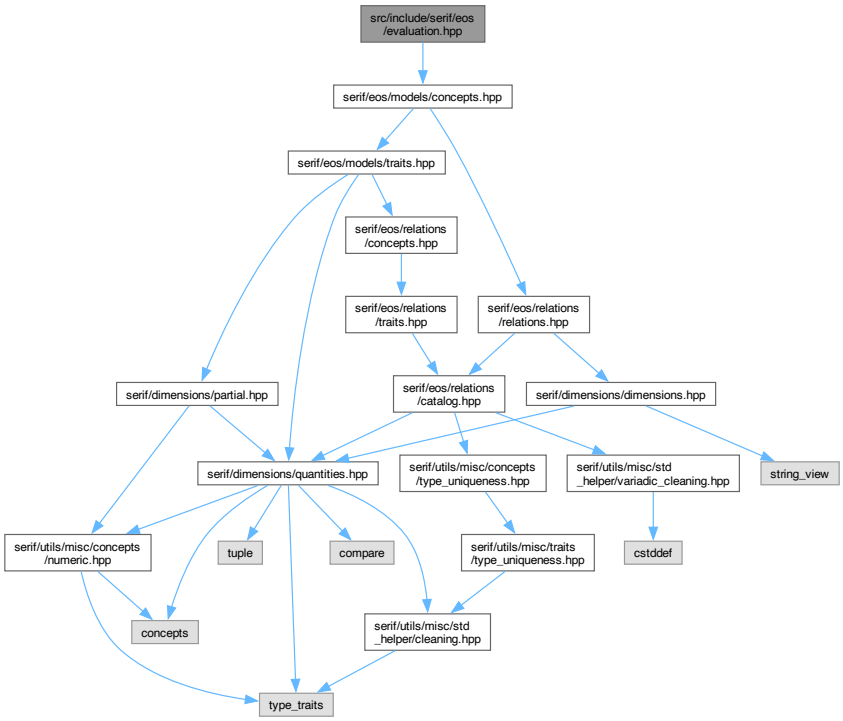
Functions

- template<typename FirstQuantityT, typename SecondQuantityT>
  consteval bool serif::eos::runtime::runtimeQuantityIdentifiersAreCompatible ()
- template<typename QuantityTupleT, std::size_t First, std::size_t... Offsets>
  consteval bool serif::eos::runtime::runtimeQuantityIdentifierIsUnambiguous (std::index_sequence< Offsets... >)
- template<typename QuantityTupleT, std::size_t... Indices>
  consteval bool serif::eos::runtime::runtimeQuantityIdentifiersAreUnambiguous (std::index_sequence< Indices... >)
- std::expected< double, utils::errors::SERiFErrorCode > serif::eos::runtime::runtimeEvaluationFailure (const utils::errors::SERiFErrorCode code)
- template<typename EOS, typename OutputT, typename... InputTs>
  std::expected< double, utils::errors::SERiFErrorCode > serif::eos::runtime::evaluateRuntimeRelation (const EOS &eos, relations::Relation< OutputT, InputTs... >, const std::span< const dimensions::runtime::RuntimeQua > inputValues)
- template<typename InputQuantityT, typename EOS, typename OutputT, typename... InputTs>
  bool serif::eos::runtime::tryRuntimePartialDerivative (const EOS &eos, relations::Relation< OutputT, InputTs... >, const dimensions::runtime::ThermodynamicQuantityID withRespectTo, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues, std::expected< double, utils::errors::SERiFErrorCode > &result)
- template<typename EOS, typename OutputT, typename... InputTs>
  std::expected< double, utils::errors::SERiFErrorCode > serif::eos::runtime::evaluateRuntimePartialDerivative (const EOS &eos, relations::Relation< OutputT, InputTs... > relation, const dimensions::runtime::ThermodynamicQuan withRespectTo, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues)
- template<typename EOS, typename RelationT>
  bool serif::eos::runtime::runtimeRelationMatches (const dimensions::runtime::ThermodynamicQuantityID outputQuantityID, const std::span< const dimensions::runtime::RuntimeQuantityValue > inputValues)

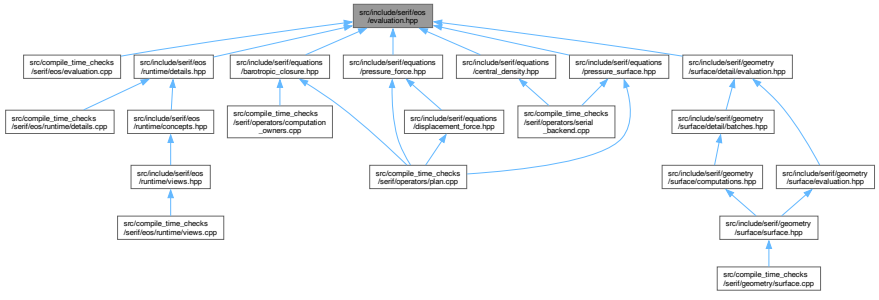
11.131 src/include/serif/geometry/surface/details.hpp File Reference

11.132 src/include/serif/eos/evaluation.hpp File Reference

#include "serif/eos/models/concepts.hpp"
Include dependency graph for evaluation.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

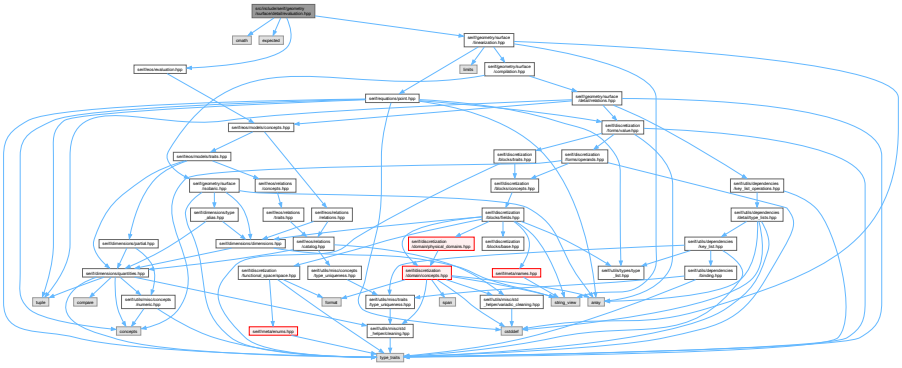
- namespace serif
- namespace serif::eos

Functions

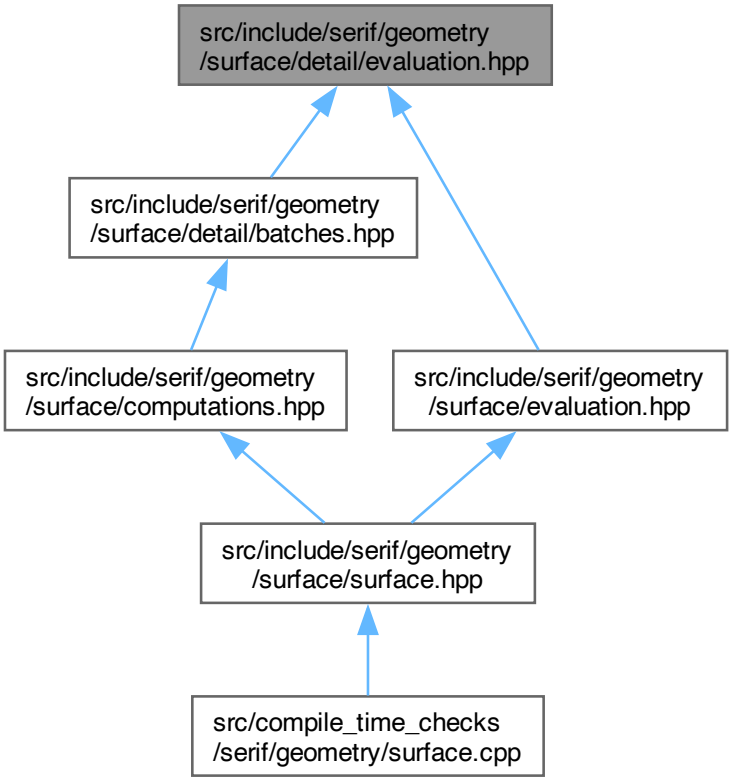
- template<dimensions::ThermodynamicQuantityType OutputQ, models::EOSModel EOS, dimensions::QuantityValue↩ Type... InputValues>
  requires models::EOSSupportingRelation<EOS, relations::Relation<OutputQ, dimensions::QuantityOfT<Input↩ Values>...>>
  constexpr dimensions::QuantityValue< OutputQ > serif::eos::evaluate (const EOS &eos, const Input↩ Values... inputValues) noexcept(noexcept(eos.evaluate(relations::Relation< OutputQ, dimensions::QuantityOfT< InputValues >... >{}, inputValues...)))
- template<dimensions::ThermodynamicQuantityType OutputQ, dimensions::ThermodynamicQuantityType InputQ, models::EOSModel EOS, dimensions::QuantityValueType... InputValues>
  requires models::EOSSupportingPartialDerivative<EOS, relations::Relation<OutputQ, dimensions::QuantityOfT<Input↩ Values>...>, InputQ>
  constexpr dimensions::PartialDerivative< OutputQ, InputQ > serif::eos::partial_derivative (const EOS &eos, const InputValues... inputValues) noexcept(noexcept(eos.partial_derivative(relations::Relation< OutputQ, dimensions::QuantityOfT< InputValues >... >{}, dimensions::WithRespectTo< InputQ >{}, inputValues...)))

11.133 src/include/serif/geometry/surface/detail/evaluation.hpp File Reference

#include <cmath>
#include <expected>
#include "serif/eos/evaluation.hpp"
#include "serif/geometry/surface/linearization.hpp"
Include dependency graph for evaluation.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::surface
- namespace serif::geometry::surface::detail

Functions

- template<typename Quantity, PressureSurfaceFormulationType Formulation>
  dimensions::QuantityValue< Quantity > serif::geometry::surface::detail::state_value (const SurfacePoint< Formulation > &point)

  Read one thermodynamic role from its bound term, preserving the EOS argument's physical type.
- template<typename Quantity, PressureSurfaceFormulationType Formulation>
  dimensions::QuantityValue< Quantity > serif::geometry::surface::detail::relation_input (const dimensions::PressureValue pressure, const SurfacePoint< Formulation > &point)

Only the carrier-resolved formulation substitutes prescribed pressure for an EOS input.

- template<PressureSurfaceFormulationType Formulation, typename... Quantities>
  bool serif::geometry::surface::detail::finite_state (const SurfacePoint< Formulation > &point, utils::dependencies::KeyList< Quantities... >)

  Validate precisely the state read by this calculation, leaving unrelated state outside its dependencies.
- template<typename Quantity, PressureSurfaceFormulationType Formulation, typename... Inputs>
  void serif::geometry::surface::detail::accumulate_partial (const typename Formulation::EquationOf↩ State &equation_of_state, const dimensions::PressureValue pressure, const SurfacePoint< Formulation > &point, SurfaceEOSResponse< Formulation > &response, dimensions::QuantityList< Inputs... >)

  Add each quantity's chain-rule contribution, including distinct roles bound to the same unknown.
- template<PressureSurfaceFormulationType Formulation, typename... Inputs, typename... StateQuantities>
  SurfaceEOSResponse< Formulation > serif::geometry::surface::detail::evaluate_relation (const typename Formulation::EquationOfState &equation_of_state, const dimensions::PressureValue pressure, const SurfacePoint< Formulation > &point, dimensions::QuantityList< Inputs... > inputs, utils::dependencies::KeyList< StateQuantities... >)

  Evaluate one selected relation and all required partials without any hidden EOS inversion.
- template<PressureSurfaceFormulationType Formulation>
  std::expected< SurfaceEOSResponse< Formulation >, SurfaceFailure > serif::geometry::surface::detail::evaluate_surfac (const typename Formulation::EquationOfState &equation_of_state, const dimensions::PressureValue pressure, const SurfacePoint< Formulation > &point)

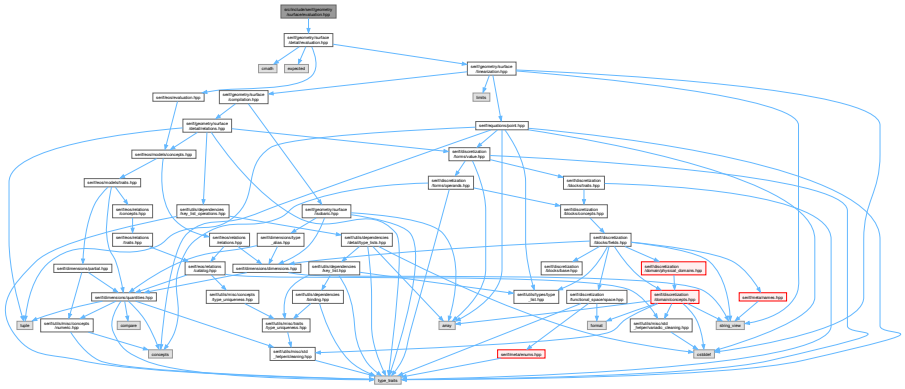
  Reject non-finite thermodynamic inputs, EOS outputs, and partials as values.
- template<PressureSurfaceFormulationType Formulation>
  SurfaceLinearization< Formulation > serif::geometry::surface::detail::linearize_surface (const dimensions::PressureValue pressure, const SurfacePoint< Formulation > &point, const SurfaceEOSResponse< Formulation > &response)

  Form the declared residual from an already evaluated EOS response, keeping the carrier identity explicit.
- template<PressureSurfaceFormulationType Formulation>
  std::expected< SurfaceLinearization< Formulation >, SurfaceFailure > serif::geometry::surface::detail::checked_lineariz (const dimensions::PressureValue pressure, const SurfacePoint< Formulation > &point, const SurfaceEOSResponse< Formulation > &response)

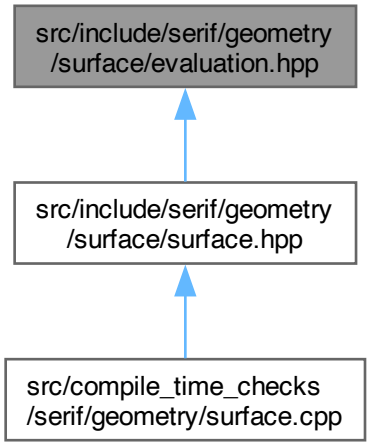
  Catch overflow in residual subtraction or derivative accumulation before publishing a linearization.

11.134 src/include/serif/geometry/surface/evaluation.hpp File Reference

#include "serif/geometry/surface/detail/evaluation.hpp"
Include dependency graph for evaluation.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::surface

Functions

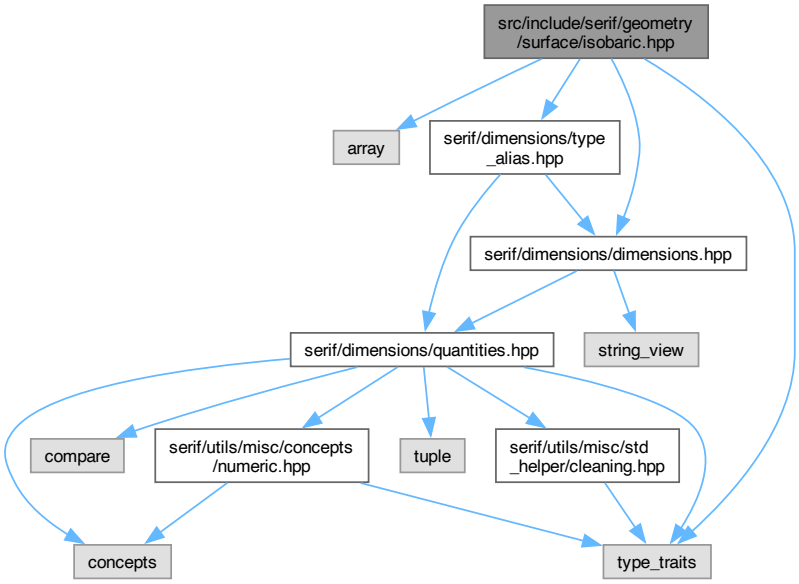
- template<PressureSurfaceFormulationType Formulation>
  std::expected< SurfaceLinearization< Formulation >, SurfaceFailure > serif::geometry::surface::evaluate_pressure_surf
  (const typename Formulation::EquationOfState &equation_of_state, const IsobaricSurface &condition, const SurfacePoint< Formulation > &point)

  Evaluate the chosen isobaric residual and its analytical derivatives at one boundary point.

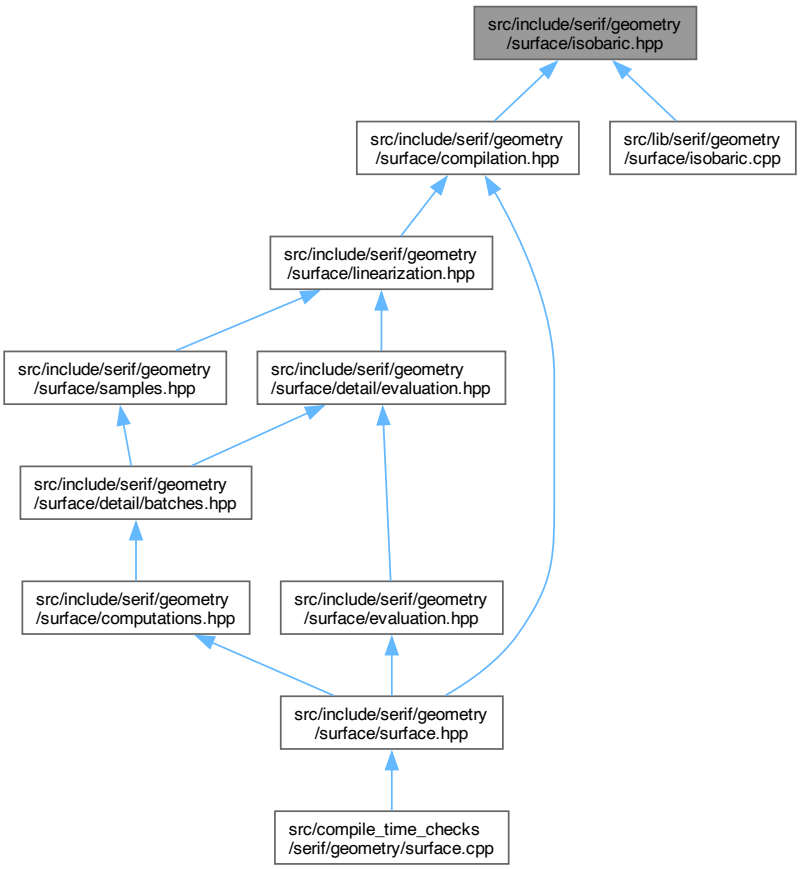
11.135 src/include/serif/geometry/surface/isobaric.hpp File Reference

#include <array>
#include <type_traits>
#include "serif/dimensions/dimensions.hpp"

#include ”serif/dimensions/type_alias.hpp”
Include dependency graph for isobaric.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::surface::PressureSurfaceDescriptor
- class serif::geometry::surface::IsobaricSurface
- struct serif::geometry::surface::IsobaricSurface::Parameters

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::surface

Concepts

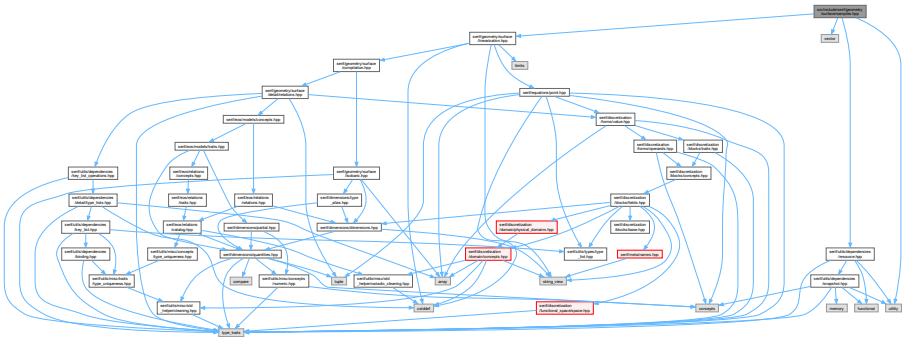
- concept serif::geometry::surface::IsobaricSurfaceType

Typedefs

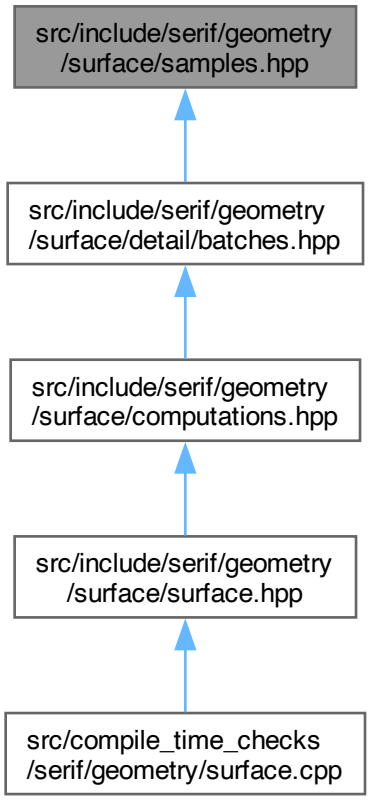
- using serif::geometry::surface::Isobaric = IsobaricSurface

11.136 src/include/serif/geometry/surface/samples.hpp File Reference

#include <utility>
#include <vector>
#include "serif/geometry/surface/linearization.hpp"
#include "serif/utils/dependencies/resource.hpp"
Include dependency graph for samples.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::geometry::surface::keys::EquationOfState
- struct serif::geometry::surface::keys::PressureCondition
- struct serif::geometry::surface::keys::EOSResponse
- struct serif::geometry::surface::keys::Linearization
- struct serif::geometry::surface::BoundaryValues< Term >

  Values of one scalar term at an ordered collection of boundary points.
- struct serif::geometry::surface::detail::BoundaryValueTracking
- struct serif::geometry::surface::detail::PressureConditionTracking

Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::surface
- namespace serif::geometry::surface::keys
- namespace serif::geometry::surface::detail

Functions

- template<SurfaceScalarTerm Term>
  auto serif::geometry::surface::boundary_values (std::vector< double > values)

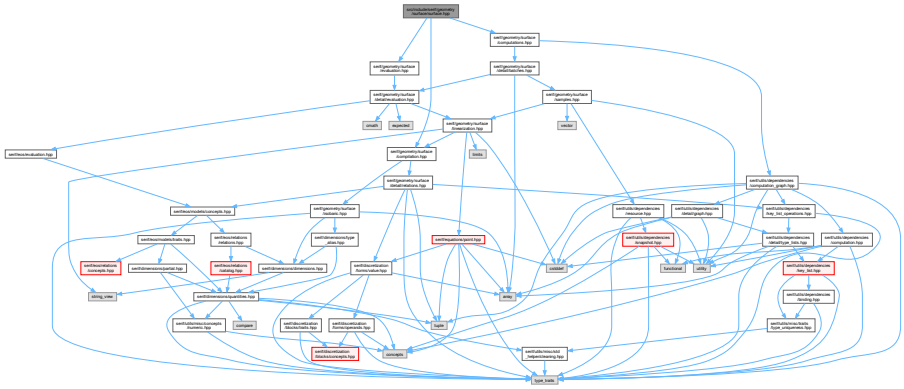
  Supply owned boundary samples to an ordinary dependency context, keyed by their physical term.
- auto serif::geometry::surface::pressure_condition (IsobaricSurface condition)

  Supply the isobaric condition as an owned dependency source, with pressure tracked by value.

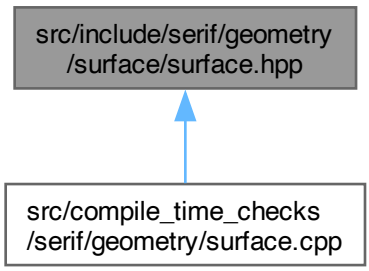
11.137 src/include/serif/geometry/surface/surface.hpp File Reference

Isobaric conditions, EOS-selected formulations, and reusable boundary calculations.

#include "serif/geometry/surface/compilation.hpp"
#include "serif/geometry/surface/computations.hpp"
#include "serif/geometry/surface/evaluation.hpp"
Include dependency graph for surface.hpp:



This graph shows which files directly or indirectly include this file:



11.137.1 Detailed Description

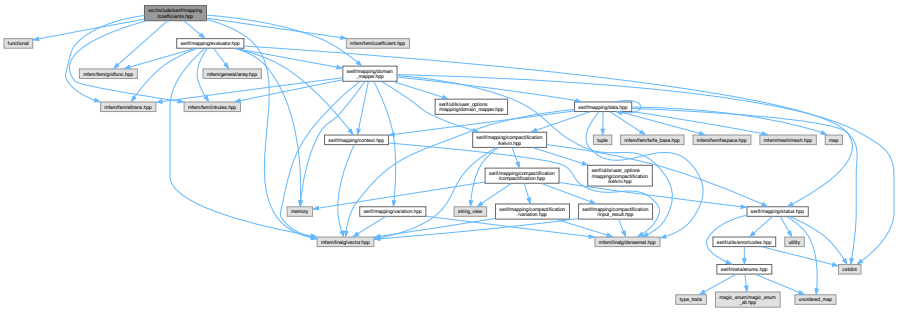
Isobaric conditions, EOS-selected formulations, and reusable boundary calculations.

IsobaricSurface describes the prescribed pressure. PressureSurfaceFormulation selects one compatible EOS relation and its carrier; DirectPressureSurfaceFormulation explicitly chooses a forward pressure residual and its equation row. Neither selects a finite-element trace.

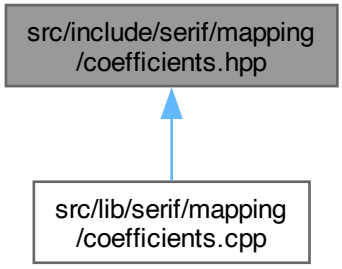
Use evaluate_pressure_surface for one sampled point, or make_pressure_surface_graph for an ordered batch. The latter returns an ordinary dependency graph: the caller supplies the EOS with its tracking policy, the pressure condition, and typed boundary values. Results use the same PointLinearization representation as other equations. Finite-element row enforcement remains a separate numerical capability.

11.138 src/include/serif/mapping/coefficients.hpp File Reference

#include <functional>
#include "mfem/fem/gridfunc.hpp"
#include "mfem/fem/coefficient.hpp"
#include "mfem/fem/eltrans.hpp"
#include "mfem/fem/intrules.hpp"
#include "mfem/linalg/vector.hpp"
#include "serif/mapping/domain_mapper.hpp"
#include "serif/mapping/evaluator.hpp"
Include dependency graph for coefficients.hpp:



This graph shows which files directly or indirectly include this file:



Classes

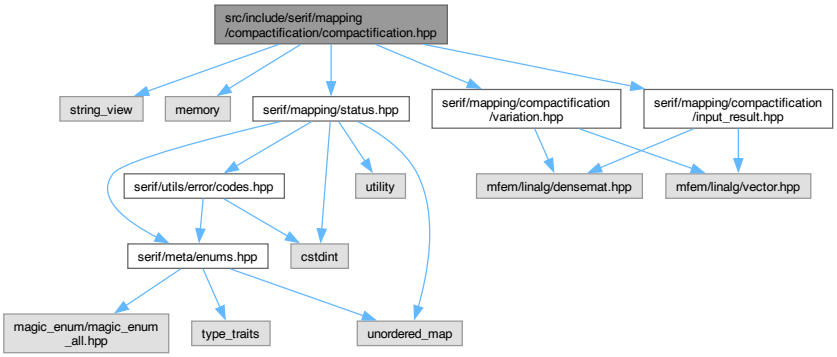
- class serif::mapping::PhysicalPositionFunctionCoefficient

Namespaces

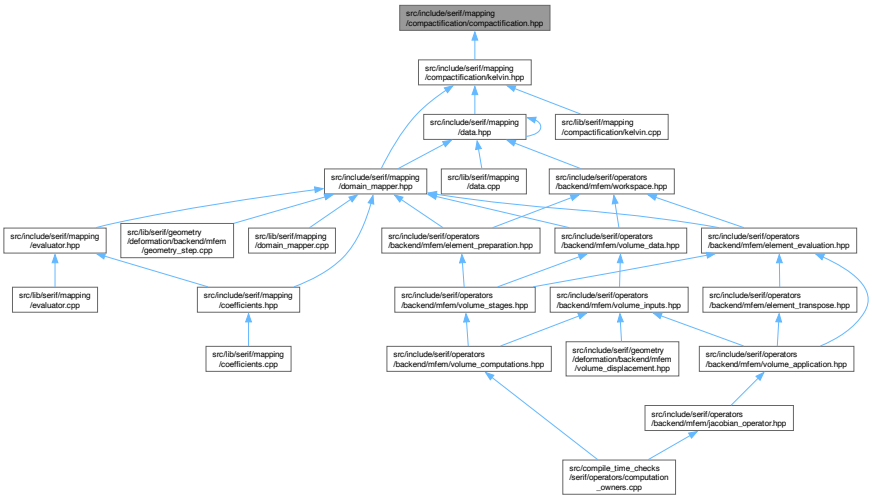
- namespace serif
- namespace serif::mapping

11.139 src/include/serif/mapping/compactification/compactification.hpp File Reference

#include <string_view>
#include <memory>
#include "serif/mapping/status.hpp"
#include "serif/mapping/compactification/input_result.hpp"
#include "serif/mapping/compactification/variation.hpp"
Include dependency graph for compactification.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::mapping::compactification::ExteriorDomainMap

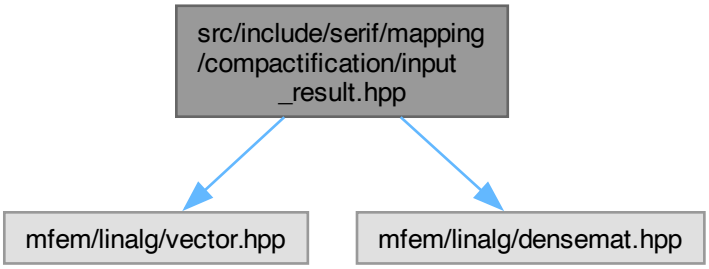
Namespaces

- namespace serif
- namespace serif::mapping
- namespace serif::mapping::compactification

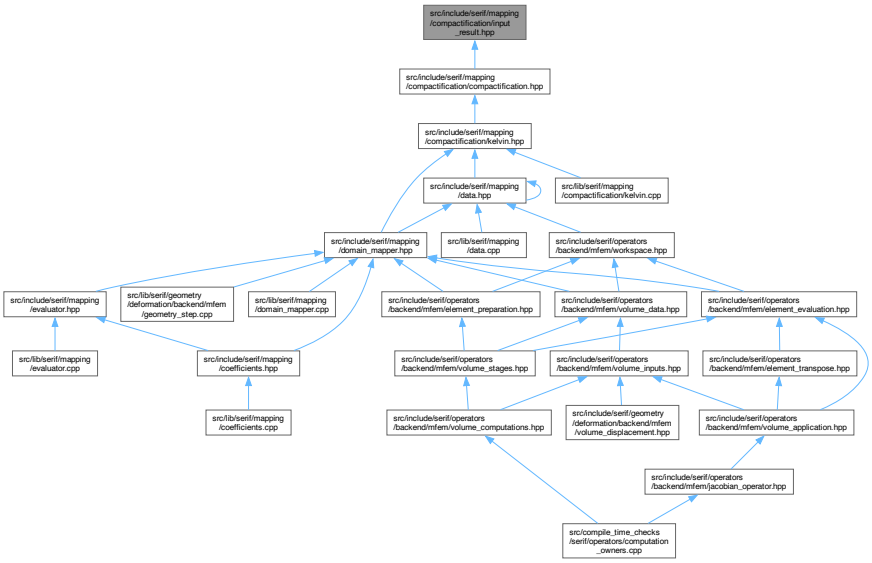
11.140 src/include/serif/mapping/compactification/input_result.hpp File Reference

#include ”mfem/linalg/vector.hpp”
#include ”mfem/linalg/densemat.hpp”

Include dependency graph for input_result.hpp:



This graph shows which files directly or indirectly include this file:



Classes

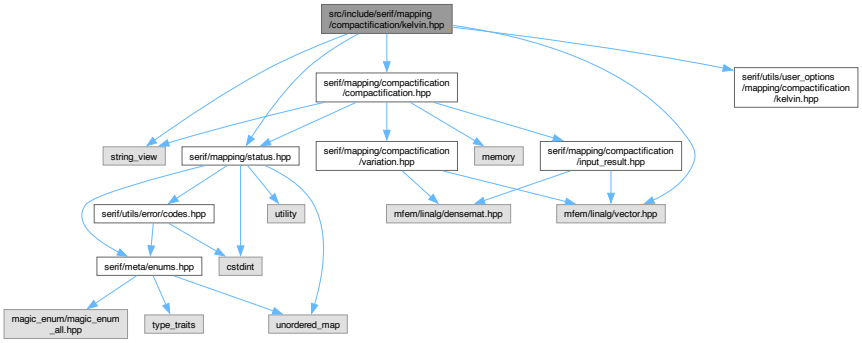
- struct serif::mapping::compactification::ExteriorMapInput
- struct serif::mapping::compactification::ExteriorMapResult

Namespaces

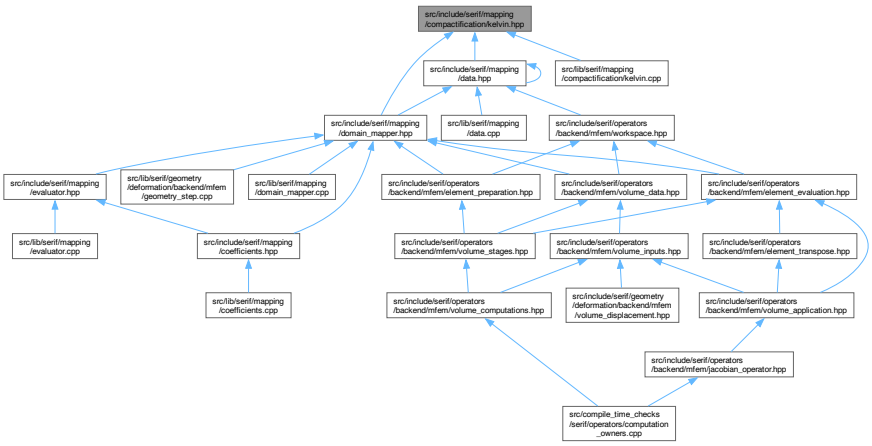
- namespace serif
- namespace serif::mapping
- namespace serif::mapping::compactification

11.141 src/include/serif/mapping/compactification/kelvin.hpp File Reference

#include <string_view>
#include "mfem/linalg/vector.hpp"
#include "serif/mapping/status.hpp"
#include "serif/mapping/compactification/compactification.hpp"
#include "serif/utils/user_options/mapping/compactification/kelvin.hpp"
Include dependency graph for kelvin.hpp:



This graph shows which files directly or indirectly include this file:



Classes

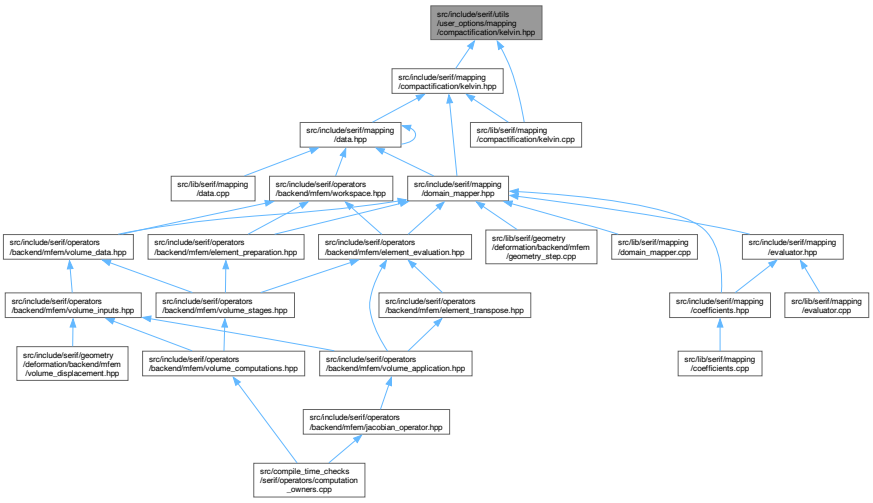
- class serif::mapping::compactification::KelvinCompactification
- struct serif::mapping::compactification::KelvinCompactification::RadialFactors

Namespaces

- namespace serif
- namespace serif::mapping
- namespace serif::mapping::compactification

11.142 src/include/serif/utils/user_↩ options/mapping/compactification/kelvin.hpp File Reference

This graph shows which files directly or indirectly include this file:



Classes

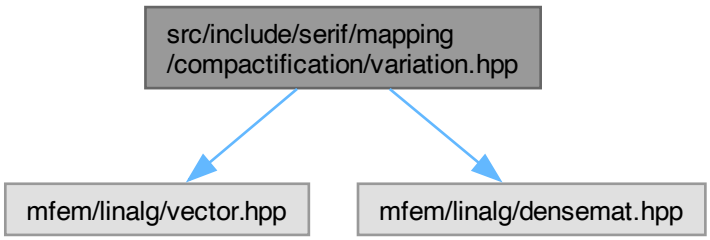
- struct serif::utils::user_options::KelvinCompactificationOptions

Namespaces

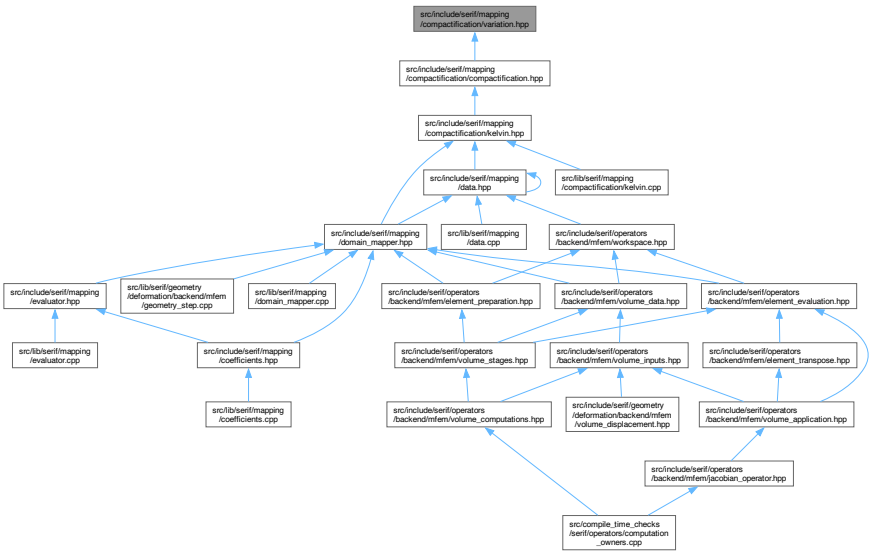
- namespace serif
- namespace serif::utils
- namespace serif::utils::user_options

11.143 src/include/serif/mapping/compactification/variation.hpp File Reference

#include ”mfem/linalg/vector.hpp”
#include ”mfem/linalg/densemat.hpp”
Include dependency graph for variation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

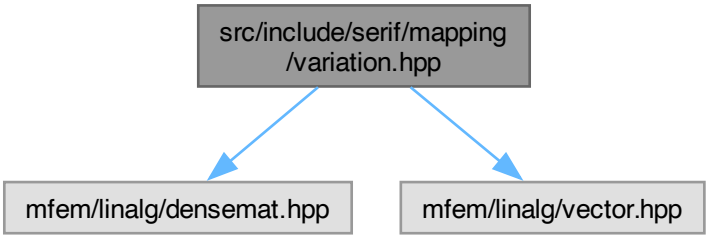
- struct serif::mapping::compactification::ExteriorMapDirection
- struct serif::mapping::compactification::ExteriorMapVariation

Namespaces

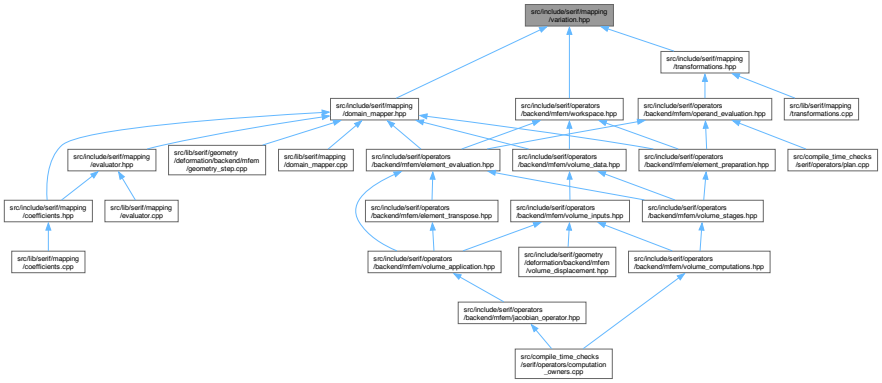
- namespace serif
- namespace serif::mapping
- namespace serif::mapping::compactification

11.144 src/include/serif/mapping/variation.hpp File Reference

#include ”mfem/linalg/densemat.hpp”
#include ”mfem/linalg/vector.hpp”
Include dependency graph for variation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

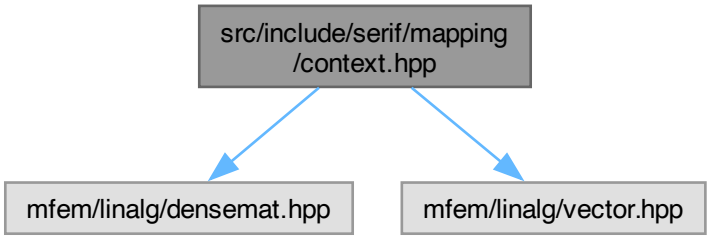
- struct serif::mapping::MappingPointVariation
- struct serif::mapping::VolumeMappingVariation
- struct serif::mapping::FaceMappingVariation

Namespaces

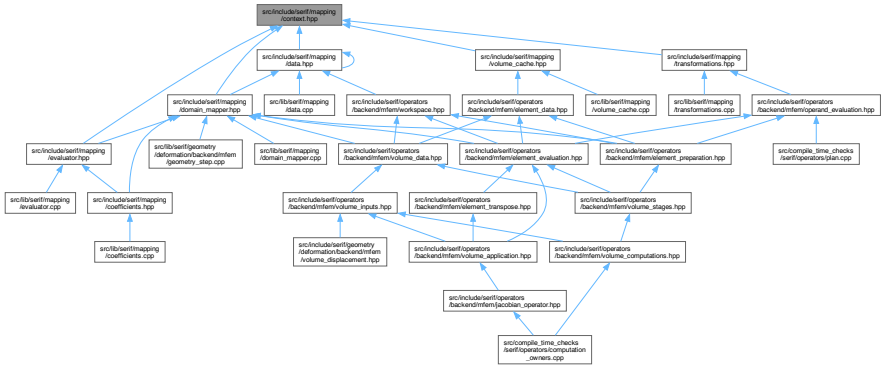
- namespace serif
- namespace serif::mapping

11.145 src/include/serif/mapping/context.hpp File Reference

#include ”mfem/linalg/densemat.hpp”
#include ”mfem/linalg/vector.hpp”
Include dependency graph for context.hpp:



This graph shows which files directly or indirectly include this file:



Classes

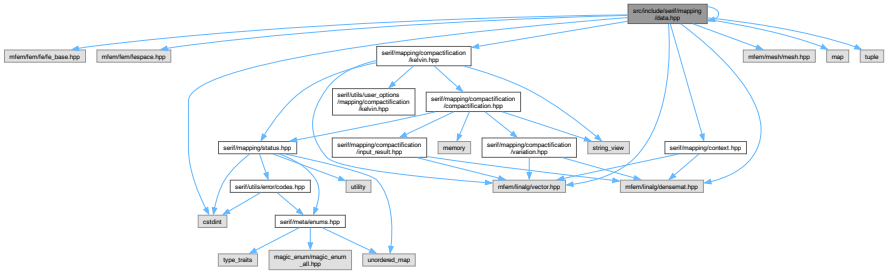
- struct serif::mapping::VolumeQuadratureContext
- struct serif::mapping::FaceQuadratureContext
- struct serif::mapping::MappingPointContext
- struct serif::mapping::VolumeMappingContext
- struct serif::mapping::FaceMappingContext

Namespaces

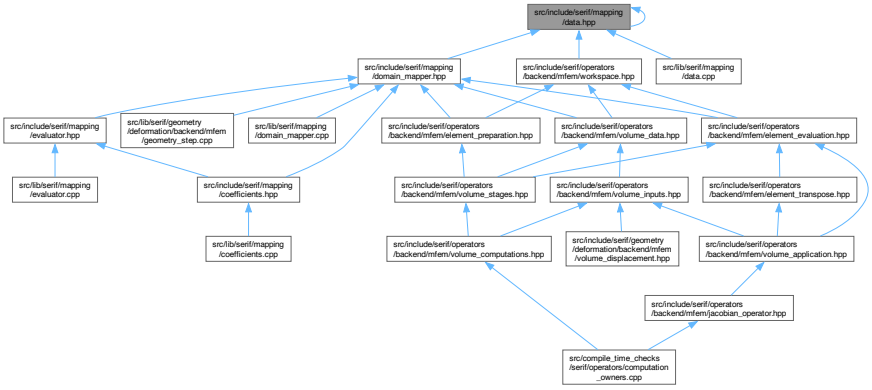
- namespace serif
- namespace serif::mapping

11.146 src/include/serif/mapping/data.hpp File Reference

#include ”mfem/fem/fe/fe_base.hpp”
#include ”mfem/fem/fespace.hpp”
#include ”mfem/linalg/vector.hpp”
#include ”mfem/linalg/densemat.hpp”
#include ”mfem/mesh/mesh.hpp”
#include <cstdint>
#include <map>
#include <tuple>
#include ”serif/mapping/compactification/kelvin.hpp”
#include ”serif/mapping/context.hpp”
#include ”serif/mapping/data.hpp”
Include dependency graph for data.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::mapping::ElementDisplacementData
- struct serif::mapping::CompactificationPointData
- class serif::mapping::ElementCompactificationData
- struct serif::mapping::ElementMappingData
- class serif::mapping::DMWorkspace
- struct serif::mapping::DMFixedMeshGeometry
- struct serif::mapping::DMGeometryCacheStatistics
- struct serif::mapping::DMReferenceDisplacementSample
- struct serif::mapping::DMGeometryCache

Namespaces

- namespace serif
- namespace serif::mapping

Enumerations

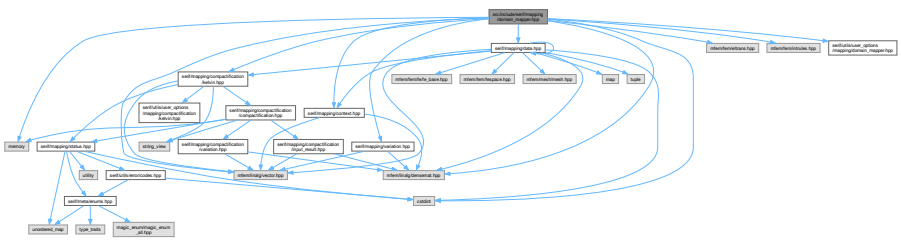
- enum class serif::mapping::FaceElementSide : uint8_t { serif::mapping::ELEMENT_1 , serif::mapping::ELEMENT_2 }

Functions

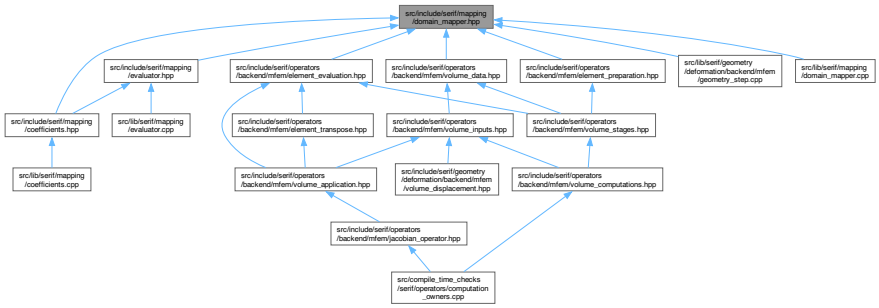
- ElementDisplacementData serif::mapping::element_displacement_data_from_element_vdofs (const mfem::FiniteElement &element, mfem::Vector &displacement_dofs)

11.147 src/include/serif/mapping/domain_mapper.hpp File Reference

#include <cstdint>
#include <memory>
#include "mfem/linalg/vector.hpp"
#include "mfem/linalg/densemat.hpp"
#include "mfem/fem/eltrans.hpp"
#include "mfem/fem/intrules.hpp"
#include "serif/mapping/compactification/kelvin.hpp"
#include "serif/mapping/context.hpp"
#include "serif/mapping/data.hpp"
#include "serif/mapping/variation.hpp"
#include "serif/utils/user_options/mapping/domain_mapper.hpp"
Include dependency graph for domain_mapper.hpp:



This graph shows which files directly or indirectly include this file:



Classes

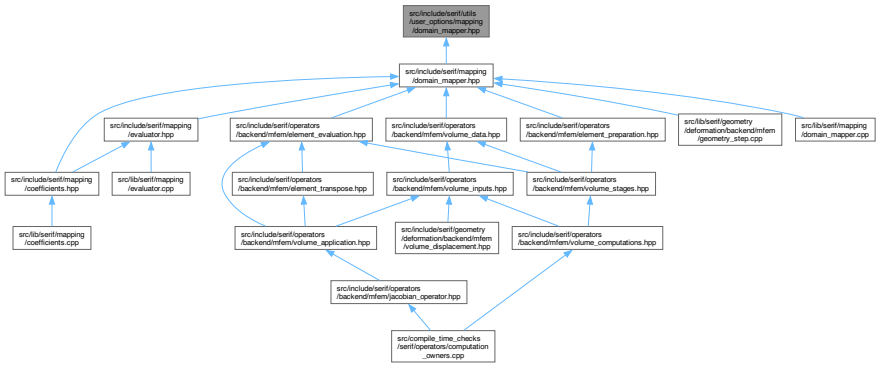
- class serif::mapping::DomainMapper

Namespaces

- namespace serif
- namespace serif::mapping

11.148 src/include/serif/utils/user_options/mapping/domain_mapper.hpp File Reference

This graph shows which files directly or indirectly include this file:



Classes

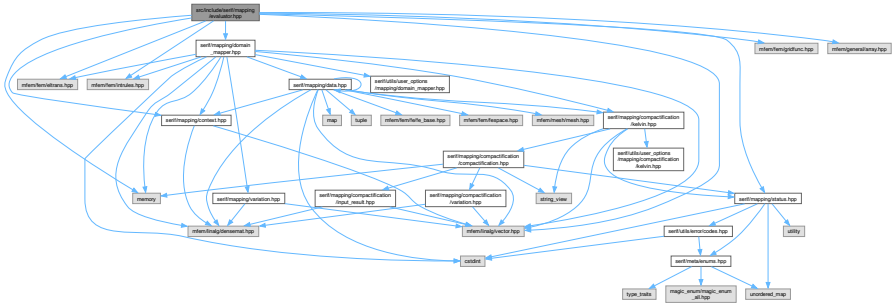
- struct serif::utils::user_options::DomainMapperOptions

Namespaces

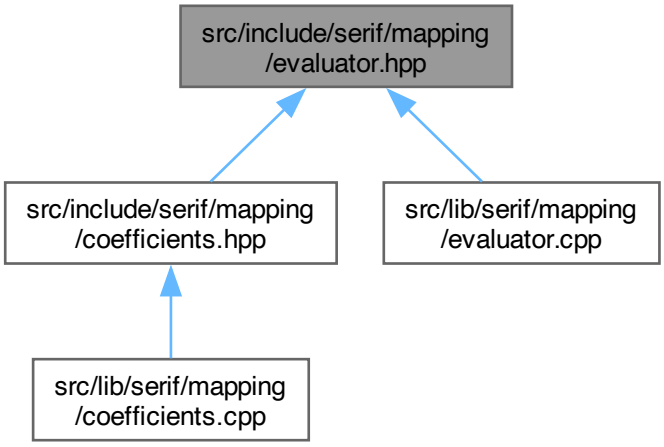
- namespace serif
- namespace serif::utils
- namespace serif::utils::user_options

11.149 src/include/serif/mapping/evaluator.hpp File Reference

#include <memory>
#include "mfem/fem/gridfunc.hpp"
#include "mfem/fem/eltrans.hpp"
#include "mfem/fem/intrules.hpp"
#include "mfem/linalg/vector.hpp"
#include "mfem/general/array.hpp"
#include "serif/mapping/domain_mapper.hpp"
#include "serif/mapping/context.hpp"
#include "serif/mapping/status.hpp"
Include dependency graph for evaluator.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::mapping::GridFunctionMappingEvaluator

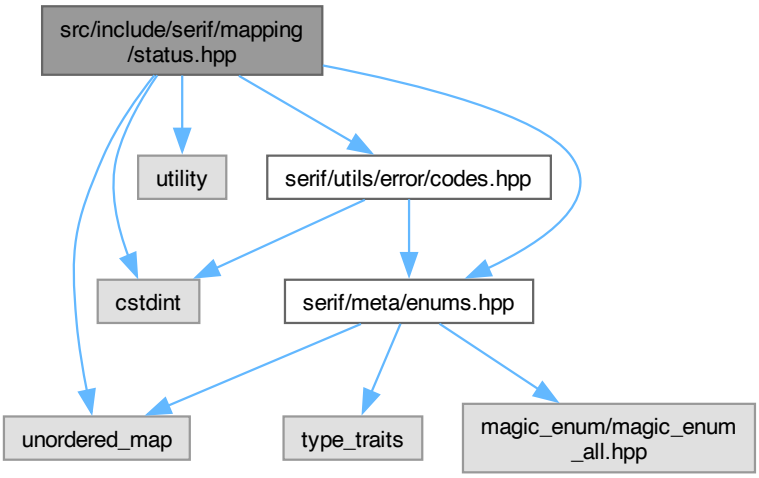
Namespaces

- namespace serif
- namespace serif::mapping

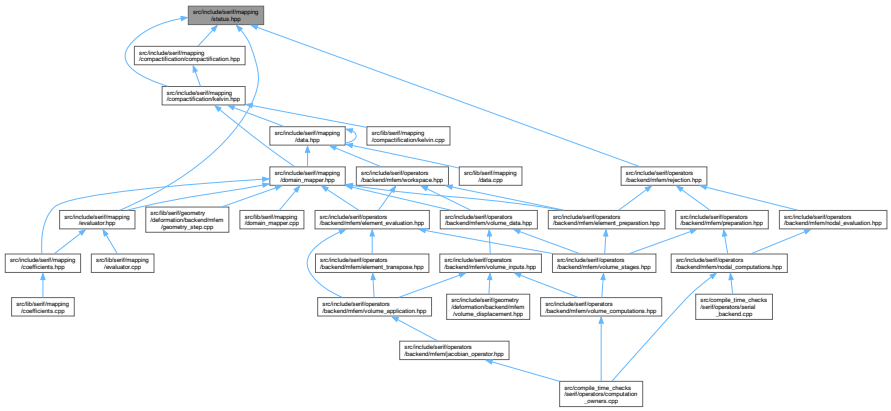
11.150 src/include/serif/mapping/status.hpp File Reference

#include <unordered_map>
#include <cstdint>
#include <utility>
#include "serif/utils/error/codes.hpp"

#include ”serif/meta/enums.hpp”
Include dependency graph for status.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::mapping

Enumerations

- enum class serif::mapping::MappingStatus : uint16_t {
  serif::mapping::VALID , serif::mapping::INVALID_DIMENSION = 400 , serif::mapping::NON_FINITE_MAPPING_I
  , serif::mapping::INVALID_REFERENCE_RADIUS ,
  serif::mapping::AT_COMPACTIFIED_INFINITY , serif::mapping::OUTSIDE_REFERENCE_DOMAIN
  , serif::mapping::NON_FINITE_MAPPING_RESULT , serif::mapping::NON_POSITIVE_DETERMINATE
  }

Functions

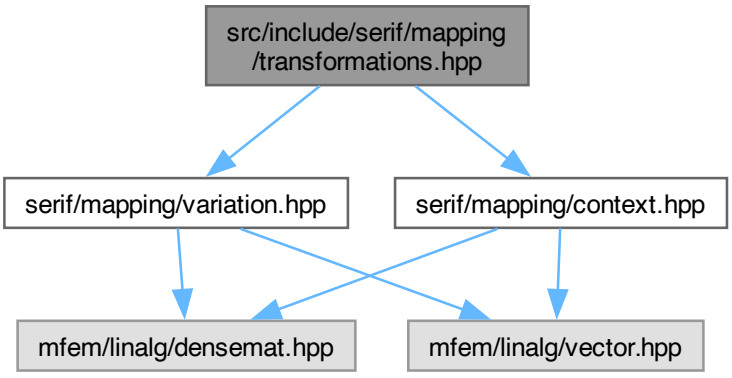
- std::unordered_map< MappingStatus, utils::errors::SERiFErrorCode > serif::mapping::make_MappingStatus_SERiFE
  ()
- std::unordered_map< utils::errors::SERiFErrorCode, MappingStatus > serif::mapping::make_SERIFErrorCode_Mapp
  ()

Variables

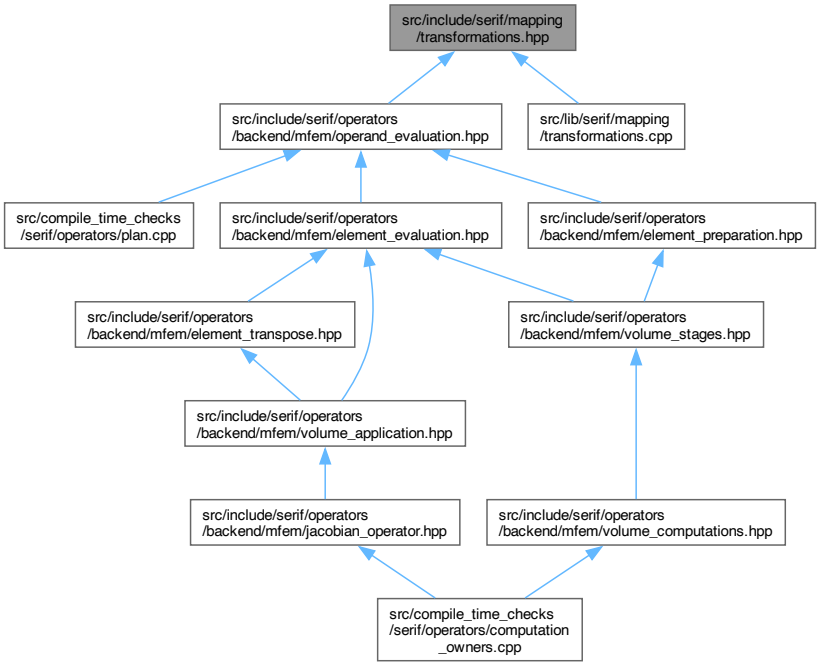
- static auto serif::mapping::MappingStatus_enum_string_map = meta::enum_to_enum_string_map<MappingStatus>
- static auto serif::mapping::MappingStatus_string_enum_map = meta::enum_to_string_enum_map<MappingStatus>
- static auto serif::mapping::MappingStatus_SERiFErrorCode_map = make_MappingStatus_SERiFErrorCode_map()
- static auto serif::mapping::SERiFErrorCode_MappingStatus_map = make_SERIFErrorCode_MappingStatus_map()

11.151 src/include/serif/mapping/transformations.hpp File Reference

#include ”serif/mapping/variation.hpp”
#include ”serif/mapping/context.hpp”
Include dependency graph for transformations.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::mapping

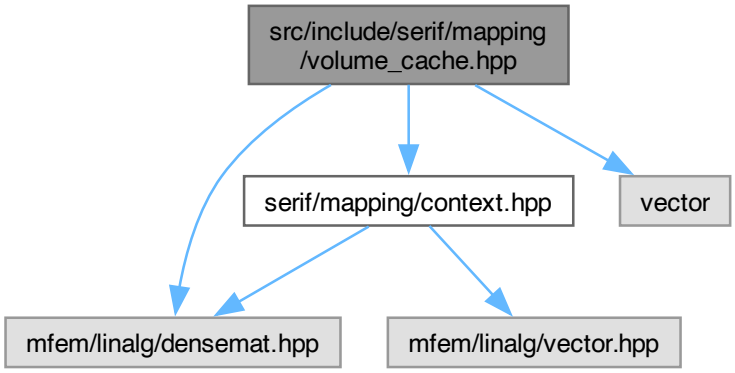
Functions

- void serif::mapping::map_hdiv_flux_to_physical (const MappingPointContext &context, const mfem::Vector &reference_flux, mfem::Vector &physical_flux)
- void serif::mapping::map_physical_flux_to_hdiv_reference (const MappingPointContext &context, const mfem::Vector &physical_flux, mfem::Vector &reference_flux)
- void serif::mapping::map_reference_gradient_to_physical (const MappingPointContext &context, const mfem::Vector &reference_gradient, mfem::Vector &physical_gradient)
- void serif::mapping::map_physical_gradient_to_reference (const MappingPointContext &context, const mfem::Vector &physical_gradient, mfem::Vector &reference_gradient)
- void serif::mapping::map_reference_vector_gradient_to_physical (const MappingPointContext &context, const mfem::DenseMatrix &reference_vector_gradient, mfem::DenseMatrix &physical_vector_↩ gradient)
- void serif::mapping::map_physical_vector_gradient_to_reference (const MappingPointContext &context, const mfem::DenseMatrix &physical_vector_gradient, mfem::DenseMatrix &reference_vector_↩ gradient)
- double serif::mapping::map_hdiv_divergence_to_physical (const MappingPointContext &context, double reference_divergence)

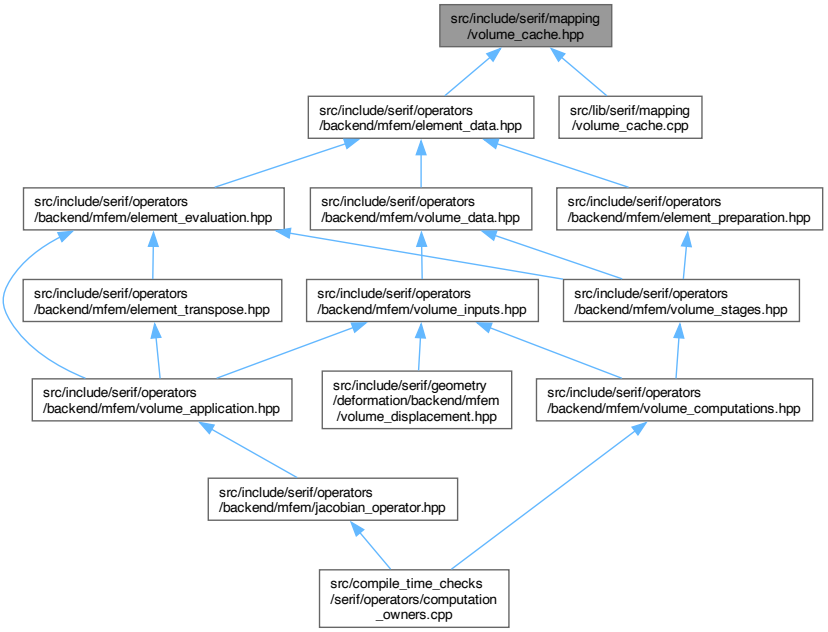
- void serif::mapping::compute_hdiv_mass_tensor (const MappingPointContext &context, mfem::↩ DenseMatrix &mass_tensor)
- void serif::mapping::compute_scalar_diffusion_tensor (const MappingPointContext &context, mfem↩ ::DenseMatrix &diffusion_tensor)
- void serif::mapping::map_hcurl_field_to_physical (const MappingPointContext &context, const mfem::Vector &reference_field, mfem::Vector &physical_field)
- void serif::mapping::map_physical_field_to_hcurl_reference (const MappingPointContext &context, const mfem::Vector &physical_field, mfem::Vector &reference_field)
- void serif::mapping::map_hcurl_curl_to_physical (const MappingPointContext &context, const mfem::Vector &reference_curl, mfem::Vector &physical_curl)
- void serif::mapping::map_physical_curl_to_hcurl_reference (const MappingPointContext &context, const mfem::Vector &physical_curl, mfem::Vector &reference_curl)
- void serif::mapping::compute_hdiv_mass_tensor_variation (const MappingPointContext &context, const MappingPointVariation &variation, mfem::DenseMatrix &mass_tensor_variation)

11.152 src/include/serif/mapping/volume_cache.hpp File Reference

#include ”mfem/linalg/densemat.hpp”
#include ”serif/mapping/context.hpp”
#include <vector>
Include dependency graph for volume_cache.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::mapping::VolumeMappingCache

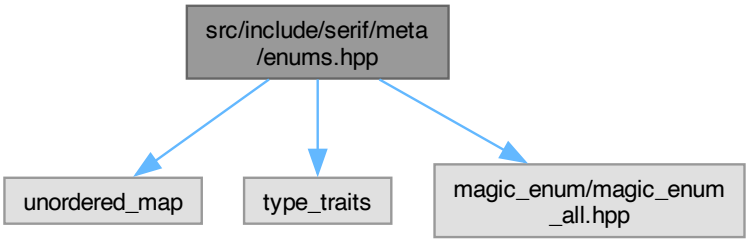
Namespaces

- namespace serif
- namespace serif::mapping

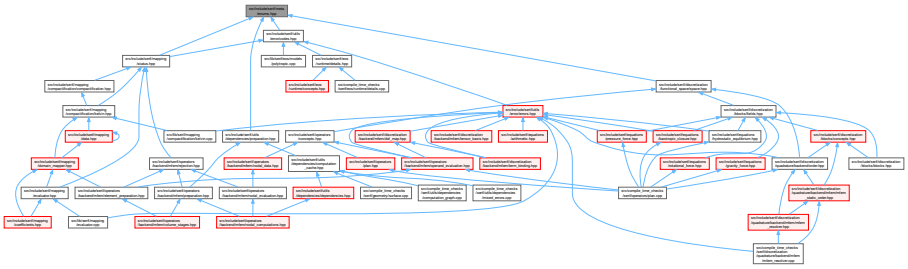
11.153 src/include/serif/meta/enums.hpp File Reference

#include <unordered_map>
#include <type_traits>

#include ”magic_enum/magic_enum_all.hpp”
Include dependency graph for enums.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

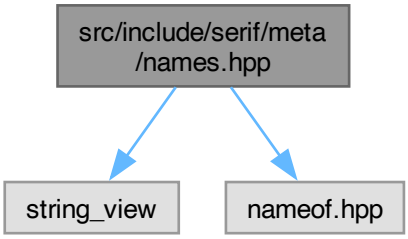
- namespace serif
- namespace serif::meta

Functions

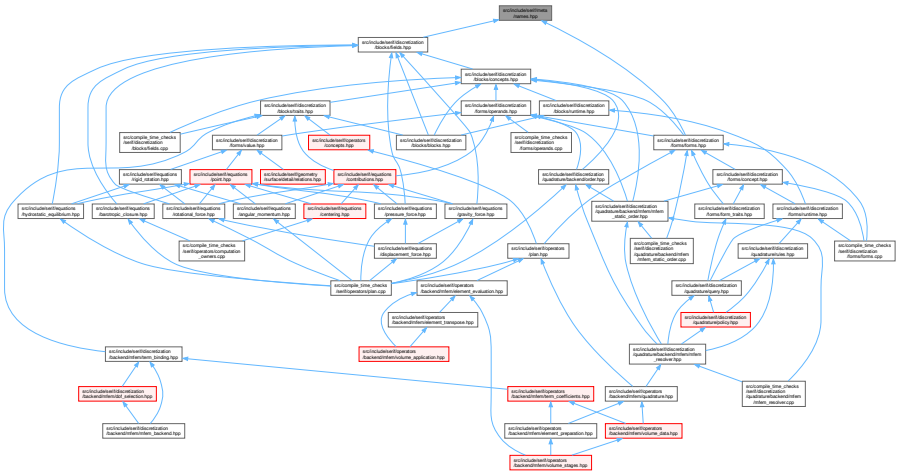
- template<typename EnumType>
  constexpr std::unordered_map< std::string_view, EnumType > serif::meta::enum_to_string_enum_map () noexcept
- template<typename EnumType>
  constexpr std::unordered_map< EnumType, std::string_view > serif::meta::enum_to_enum_string_map () noexcept
- template<typename EnumType>
  consteval std::string_view serif::meta::enum_to_string (EnumType value) noexcept

11.154 src/include/serif/meta/names.hpp File Reference

#include <string_view>
#include "nameof.hpp"
Include dependency graph for names.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

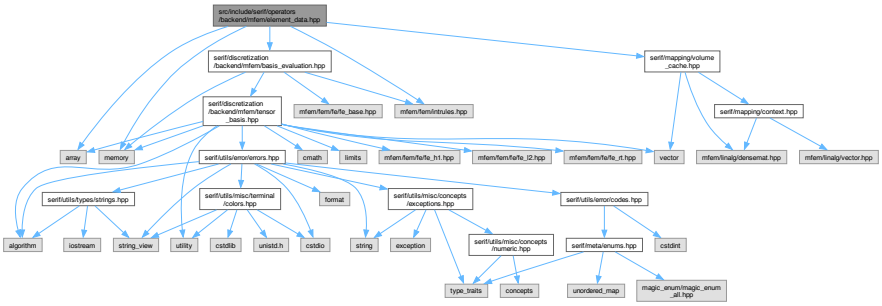
- namespace serif
- namespace serif::meta

Functions

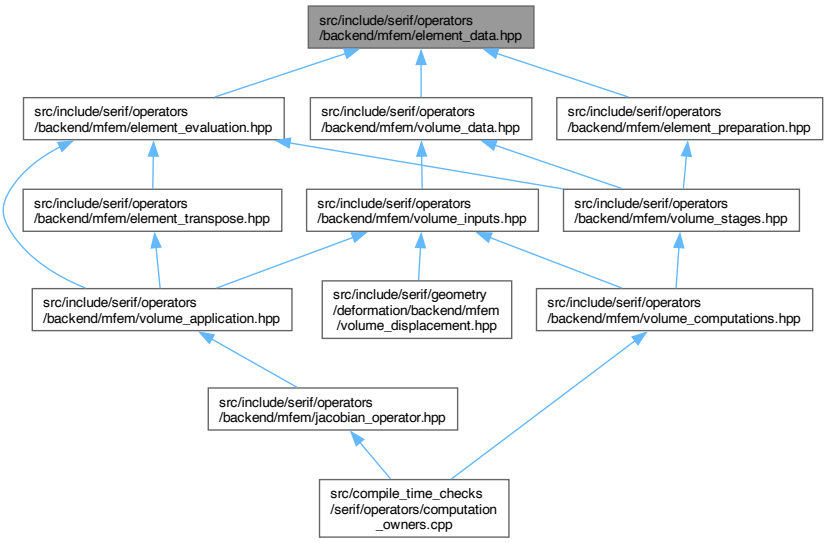
- template<typename T>
  consteval std::string_view serif::meta::type_name () noexcept
- template<typename T>
  consteval std::string_view serif::meta::template_name () noexcept

11.155 src/include/serif/operators/backend/mfem/element_data.hpp File Reference

#include <array>
#include <memory>
#include "mfem/fem/intrules.hpp"
#include "serif/mapping/volume_cache.hpp"
#include "serif/discretization/backend/mfem/basis_evaluation.hpp"
Include dependency graph for element_data.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::ElementBasis< OperandCount >

Fixed-mesh basis samples; independent of the current coefficient values and displacement.

- struct serif::operators::backend::mfem::ElementGeometry< OperandCount >

  Physical mapping and mapped bases for one frozen displacement.

- struct serif::operators::backend::mfem::ElementData< OperandCount >

  Weighted equation data retains the exact basis, geometry, and samples used to compute it.

- class serif::operators::backend::mfem::BasisEvaluation

  Represents an element-local basis evaluation operator, supporting dense or tensor contractions, pointwise mappings, and transpose application.

Namespaces

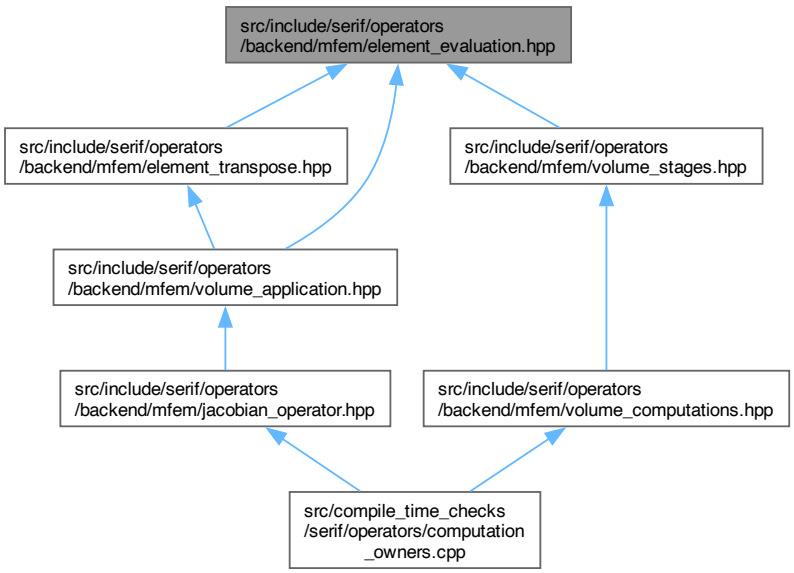
- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

11.156 src/include/serif/operators/backend/mfem/element_↩ evaluation.hpp File Reference

#include <array>
#include <cmath>
#include "serif/operators/backend/mfem/operand_evaluation.hpp"
#include "serif/operators/backend/mfem/element_data.hpp"
#include "serif/operators/backend/mfem/workspace.hpp"
#include "serif/operators/plan.hpp"
#include "serif/mapping/domain_mapper.hpp"
Include dependency graph for element_evaluation.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

- template<typename ContributionT, std::size_t Count>
  void serif::operators::backend::mfem::element_residual (const ElementData< Count > &data, ::mfem::Vector &result, BasisWorkspace &workspace)

  Integrate the cached scalar/vector physics against its actual test operation.
- template<VolumeEquation EquationT, typename ContributionT>
  auto serif::operators::backend::mfem::geometry_integrand_variation (const ElementData< EquationPlan< EquationT >::operand_count > &data, int point, const mapping::VolumeMappingContext &context, const mapping::VolumeMappingVariation &variation)

  Geometry-only variation of the weighted physics; shared by forward and transpose actions.
- template<VolumeEquation EquationT, typename ContributionT>
  void serif::operators::backend::mfem::element_jacobian_action (const ElementData< EquationPlan< EquationT >::operand_count > &data, const std::array< const discretization::backend::mfem::DofMap *, EquationPlan< EquationT >::column_count > &columns, const std::array< const ::mfem::Vector *, EquationPlan< EquationT >::column_count > &directions, const discretization::backend::mfem::DofMap &displacement_binding, const ::mfem::Vector &displacement_direction, const ::mfem::FiniteElement↩ Space *compactification_space, const mapping::DomainMapper &mapper, Workspace &workspace, ::mfem::Vector &result)

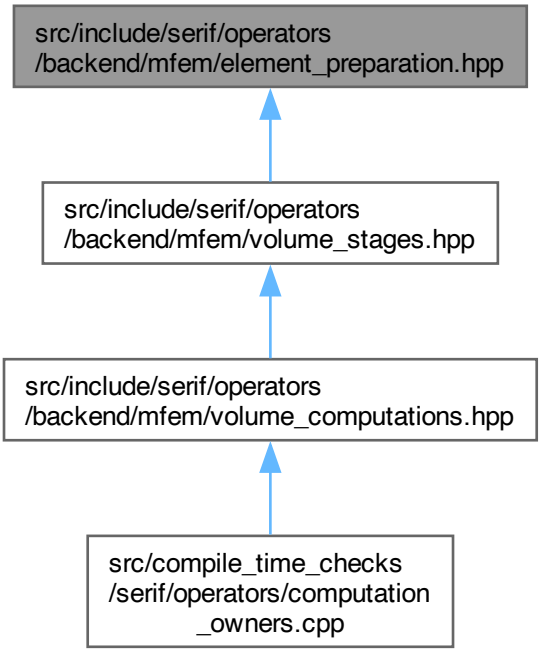
  Differentiate weight, physics, and both operand and test mapping rules in one path.

11.157 src/include/serif/operators/backend/mfem/element_preparation.hpp File Reference

#include <array>
#include <cmath>
#include <expected>
#include <memory>
#include <utility>
#include "serif/mapping/domain_mapper.hpp"
#include "serif/operators/backend/mfem/element_data.hpp"
#include "serif/operators/backend/mfem/operand_evaluation.hpp"
#include "serif/operators/backend/mfem/quadrature.hpp"
#include "serif/operators/backend/mfem/rejection.hpp"
#include "serif/operators/backend/mfem/term_coefficients.hpp"
#include "serif/operators/backend/mfem/workspace.hpp"
#include "serif/utils/misc/compensated_sum.hpp"
Include dependency graph for element_preparation.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

- template<VolumeEquation EquationT, typename ContributionT>
  ElementBasis< EquationPlan< EquationT >::operand_count > serif::operators::backend::mfem::sample_element_basis
  (const ContributionT &equation, int element, const discretization::backend::mfem::DofMap &row, const discretization::backend::mfem::DofMap &displacement_binding, const std::array< const discretization::backend::mfem::DofMap *, EquationPlan< EquationT >::column_count > &columns, const mapping::DomainMapper &mapper, const discretization::quadrature::Policy &policy, TensorBasisCache &basis_cache, ContractionMode contraction_mode)

  Sample the contribution's quadrature rule and bases on the fixed computational mesh.
- input binding serif::operators::backend::mfem::element_values (basis.element, input.values, workspace.↩ element_values)

- template<VolumeEquation EquationT, typename ContributionT>
  std::expected< ElementGeometry< EquationPlan< EquationT >::operand_count >, PreparationRejection > serif::operators::backend::mfem::map_element_geometry (const ElementBasis< EquationPlan< EquationT >::operand_count > &basis, const discretization::backend::mfem::DofMap &displacement_binding, const ::mfem::Vector &displacement, const MappingCoefficients &compactification, const mapping::DomainMapper &mapper, Workspace &workspace)

  Freeze displacement and compactification, then map quadrature points and bases to physical space.
- template<EvaluableOperand OperandT, std::size_t Count>
  ::mfem::Vector serif::operators::backend::mfem::map_element_operand (const ::mfem::Vector &reference, const ElementGeometry< Count > &geometry, Workspace &workspace)

  Apply an operand's physical transformation to its cached fixed-mesh samples.
- template<VolumeEquation EquationT, typename ContributionT>
  equations::EquationPoint< typename ContributionT::Operands > serif::operators::backend::mfem::element_equation_p (const std::array< std::shared_ptr< const ::mfem::Vector >, EquationPlan< EquationT >::operand_count > &physical_values, const mapping::MappingPointContext &context, int point)

  Pack the physical samples requested by this contribution at one quadrature point.
- template<VolumeEquation EquationT, typename ContributionT>
  bool serif::operators::backend::mfem::store_element_linearization (const equations::PointLinearization< typename ContributionT::Operands, typename OperandEvaluator< equations::TestOperandOf< ContributionT > >::Value > &physics, int point, double weight, ElementData< EquationPlan< EquationT >::operand_count > &data)

  Weight a point's residual and derivatives; reject any non-finite stored component.
- template<VolumeEquation EquationT, typename ContributionT>
  std::expected< ElementData< EquationPlan< EquationT >::operand_count >, PreparationRejection > serif::operators::backend::mfem::evaluate_element_equation (const ContributionT &equation, double coefficient, std::shared_ptr< const ElementBasis< EquationPlan< EquationT >::operand_count > > basis, std::shared_ptr< const ElementGeometry< EquationPlan< EquationT >::operand_count > > geometry, std::array< std::shared_ptr< const ::mfem::Vector >, EquationPlan< EquationT >::operand_count > reference_values, const std::array< std::shared_ptr< const ::mfem::Vector >, EquationPlan< EquationT >::operand_count > &physical_values, Workspace &workspace)

  Evaluate and weight the equation while retaining its immutable preparation inputs.

Variables

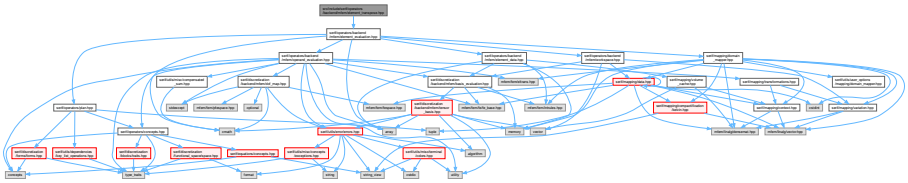
- template<VolumeEquation EquationT, EvaluableOperand OperandT, typename BindingT>
  constexpr std::size_t serif::operators::backend::mfem::index = equations::OperandValues<typename Plan::Operands>::template index<OperandT>()

  Contract one operand's fixed-mesh basis with its current coefficients.
- ::mfem::Vector serif::operators::backend::mfem::values

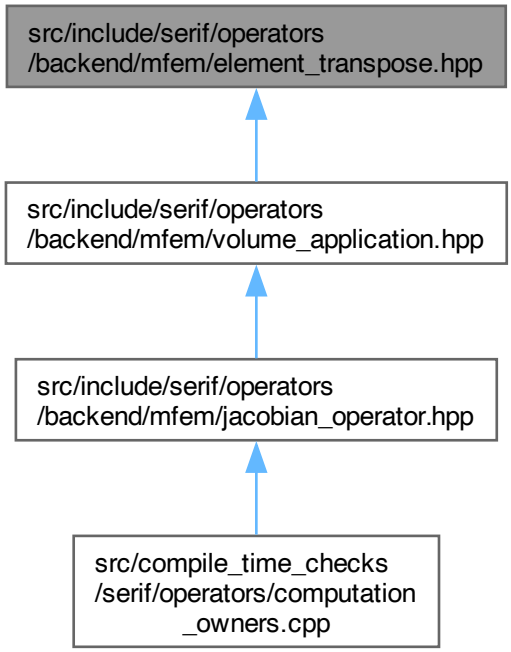
11.158 src/include/serif/operators/backend/mfem/element_transpose.hpp File Reference

#include "serif/operators/backend/mfem/element_evaluation.hpp"

Include dependency graph for element_transpose.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

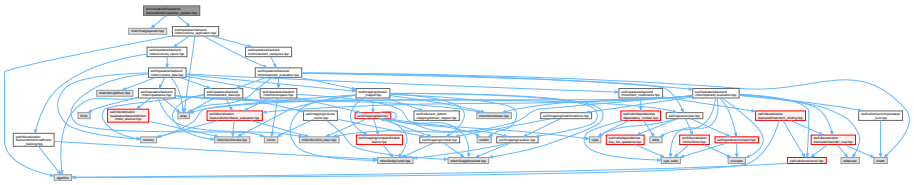
- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

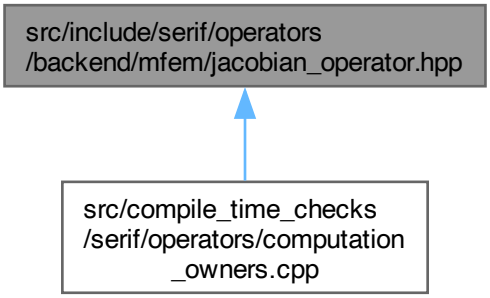
- template<VolumeEquation EquationT, typename ContributionT>
  void serif::operators::backend::mfem::element_jacobian_transpose (const ElementData< EquationPlan< EquationT >::operand_count > &data, const discretization::backend::mfem::DofMap &row, const discretization::backend::mfem::DofMap &displacement_binding, const ::mfem::Vector &residual_dual, const ::mfem::FiniteElementSpace &compactification_space, const mapping::DomainMapper &mapper, Workspace &workspace, std::array<::mfem::Vector, EquationPlan< EquationT >::column_count > &column_duals, ::mfem::Vector &displacement_dual)

11.159 src/include/serif/operators/backend/mfem/jacobian_operator.hpp File Reference

#include "mfem/linalg/operator.hpp"
#include "serif/operators/backend/mfem/volume_application.hpp"
Include dependency graph for jacobian_operator.hpp:



This graph shows which files directly or indirectly include this file:



Classes

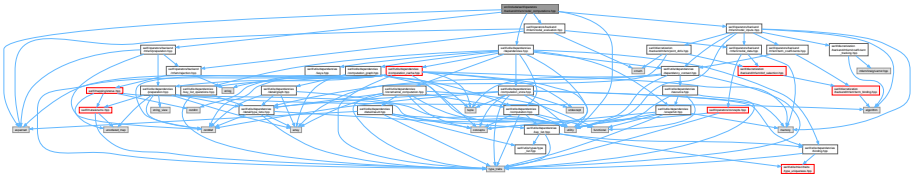
- class serif::operators::backend::mfem::JacobianOperator< CacheT >

Namespaces

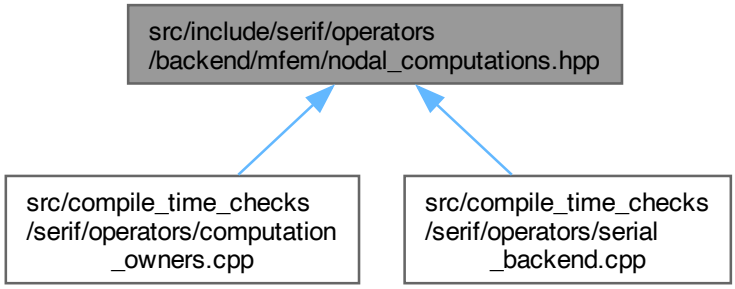
- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

11.160 src/include/serif/operators/backend/mfem/nodal_↩ computations.hpp File Reference

#include <expected>
#include <utility>
#include "serif/operators/backend/mfem/nodal_evaluation.hpp"
#include "serif/operators/backend/mfem/nodal_inputs.hpp"
#include "serif/operators/backend/mfem/preparation.hpp"
#include "serif/utils/dependencies/dependencies.hpp"
Include dependency graph for nodal_computations.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  auto serif::operators::backend::mfem::make_nodal_equation_computation (EquationT equation, NodalDiscretization< EquationT, SchemaT > discretization)

  Declare the point or boundary calculation; its fixed equation and selection move with the graph.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  auto serif::operators::backend::mfem::make_nodal_equation_cache (EquationT equation, const NodalDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  Construct an ordinary dependency cache without evaluating the nodal equation.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  auto serif::operators::backend::mfem::make_point_border_cache (EquationT equation, const PointBorderDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  Keep the border scalar and point linearization in one dependency graph and one accepted result.
- template<typename CacheT, NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  std::expected< void, PreparationRejection > serif::operators::backend::mfem::try_prepare_nodal (CacheT &cache, const NodalDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients, const PreparationMode mode=PreparationMode::reuse_unchanged)

  Validate and gather the selected coefficients, then let the dependency cache decide what needs evaluation.
- template<typename CacheT, AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  std::expected< void, PreparationRejection > serif::operators::backend::mfem::try_prepare_point_border (CacheT &cache, const PointBorderDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients, const PreparationMode mode=PreparationMode::reuse_unchanged)

  Validate both point and scalar sources before publishing the complete border calculation.
- template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
  void serif::operators::backend::mfem::prepare_nodal (CacheT &cache, const DiscretizationT &discretization, const CoefficientsT &term_coefficients, const PreparationMode mode=PreparationMode::reuse_unchanged)

  Translate a numerical rejection at the caller-facing throwing boundary.
- template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
  void serif::operators::backend::mfem::prepare_point_border (CacheT &cache, const DiscretizationT &discretization, const CoefficientsT &term_coefficients, const PreparationMode mode=PreparationMode::reuse_unchanged)

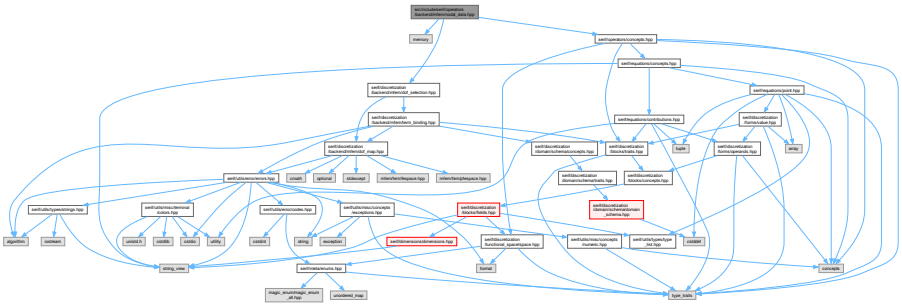
  Translate an affine border's numerical rejection at the caller-facing throwing boundary.
- template<typename CacheT>
  NumericalWork serif::operators::backend::mfem::nodal_work_report (const CacheT &cache)

  Derive numerical work from the dependency report, without a second report lifecycle.

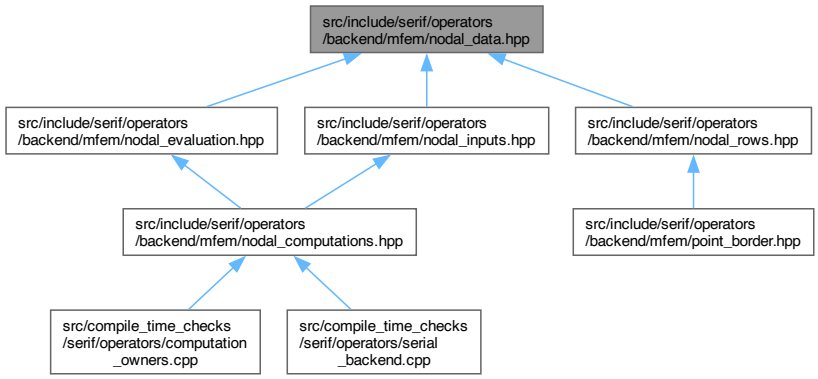
11.161 src/include/serif/operators/backend/mfem/nodal_data.hpp File Reference

#include <memory>
#include "serif/discretization/backend/mfem/dof_selection.hpp"

#include ”serif/operators/concepts.hpp”
Include dependency graph for nodal_data.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >
  Fixed coefficient ordering and selected rows shared by the graph and its accepted results.
- struct serif::operators::backend::mfem::NodalData< EquationT, SchemaT >
  A frozen nodal residual and derivative retain the selection needed to interpret every entry.
- struct serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >
  A point condition and its scalar border must use the same mesh.
- struct serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >
  Store the scalar and point linearization together so a result cannot mix accepted states.

Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

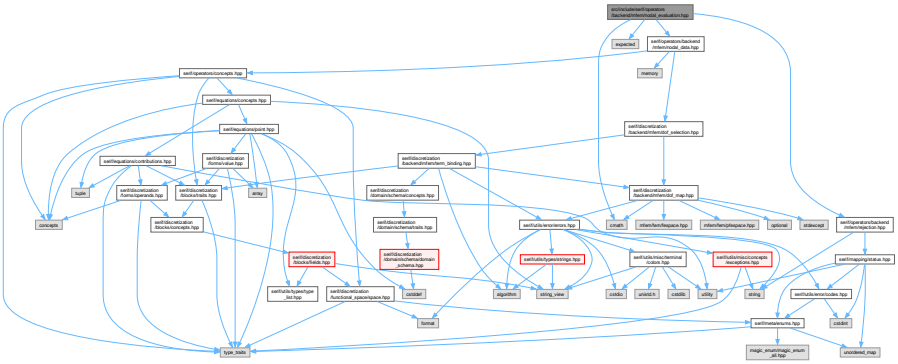
Concepts

- concept serif::operators::backend::mfem::AffinePointEquation

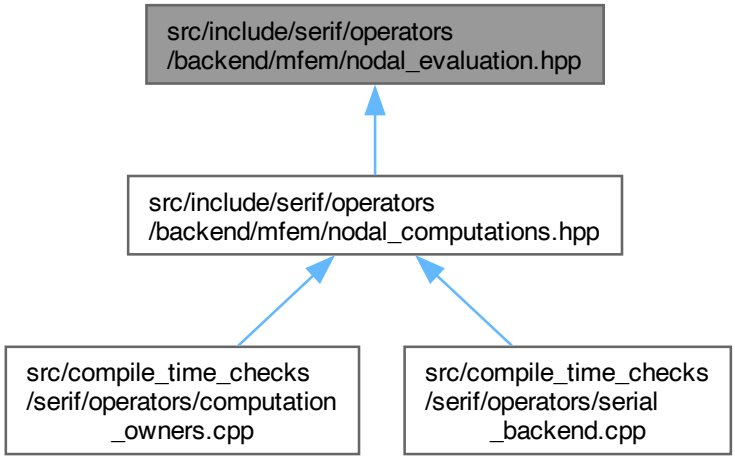
  An affine scalar point condition supplies the two off-diagonal border blocks.

11.162 src/include/serif/operators/backend/mfem/nodal_evaluation.hpp File Reference

#include <cmath>
#include <expected>
#include "serif/operators/backend/mfem/nodal_data.hpp"
#include "serif/operators/backend/mfem/rejection.hpp"
Include dependency graph for nodal_evaluation.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

- template<NodalValueEquation EquationT>
  equations::EquationPoint< typename EquationT::Operands > serif::operators::backend::mfem::make_nodal_equation_
  (const ::mfem::Vector &values, const int first)

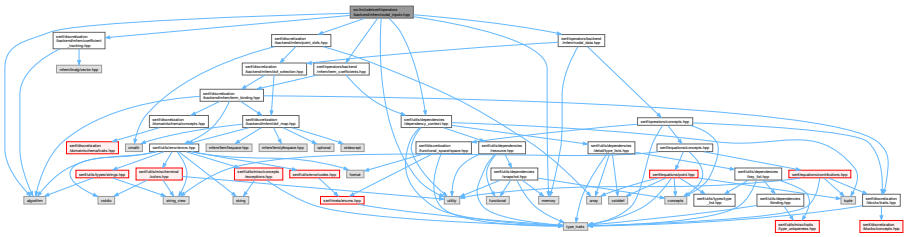
  Convert one scalar node or one complete vector point into the equation's declared input.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  std::expected< void, PreparationRejection > serif::operators::backend::mfem::store_nodal_linearization
  (const equations::PointLinearization< typename EquationT::Operands, discretization::forms::OperandValue<
  equations::TestOperandOf< EquationT > > > &physics, const int first, NodalData< EquationT,
  SchemaT > &data)

  Accept only finite, position-independent linearizations before exposing their row data.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  std::expected< NodalData< EquationT, SchemaT >, PreparationRejection > serif::operators::backend::mfem::evaluate_
  (const EquationT &equation, const NodalDiscretization< EquationT, SchemaT > &discretization,
  const ::mfem::Vector &values)

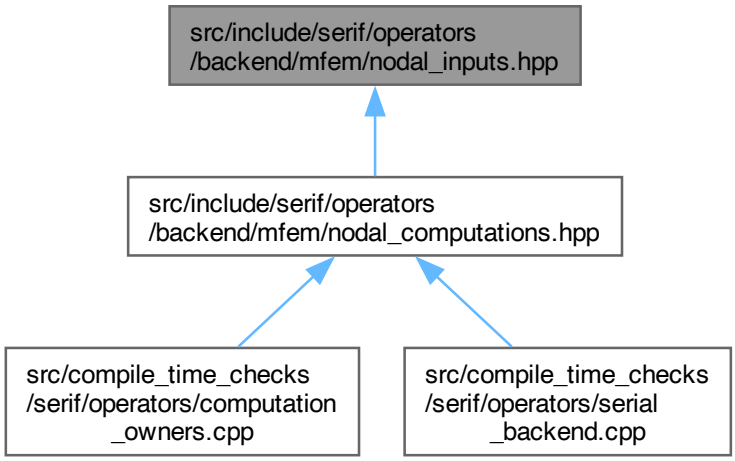
  Evaluate the equation once per selected point, keeping residuals and derivatives in one cached result.

11.163 src/include/serif/operators/backend/mfem/nodal_inputs.hpp File Reference

#include <algorithm>
#include <memory>
#include <utility>
#include "serif/discretization/backend/mfem/coefficient_tracking.hpp"
#include "serif/discretization/backend/mfem/point_dofs.hpp"
#include "serif/operators/backend/mfem/nodal_data.hpp"
#include "serif/operators/backend/mfem/term_coefficients.hpp"
#include "serif/utils/dependencies/dependency_context.hpp"
Include dependency graph for nodal_inputs.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::nodal_keys::Linearization
- struct serif::operators::backend::mfem::nodal_keys::BorderLinearization
- struct serif::operators::backend::mfem::SelectedNodalValues

  Selected coefficients are owned by the context; the selection has a stable shared identity.
- struct serif::operators::backend::mfem::NodalValueTracking

  Moving the context or copying equal coefficients does not invalidate the nodal calculation.

Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem
- namespace serif::operators::backend::mfem::nodal_keys

Functions

- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void serif::operators::backend::mfem::validate_nodal_selection (const NodalDiscretization< EquationT, SchemaT > &discretization)

  Check stored selection invariants before dereferencing or gathering any coefficient.

- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void serif::operators::backend::mfem::validate_nodal_discretization (const NodalDiscretization< EquationT, SchemaT > &discretization)

  Admit only nodal spaces and the complete selection required by the equation's point or boundary support.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  auto serif::operators::backend::mfem::make_nodal_discretization (const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > &residual_binding, discretization::backend::mfem::DofSelection selected_rows)

  Validate a caller-supplied selection immediately, before constructing its computation graph.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  requires requires { EquationT::reference_position; }
  auto serif::operators::backend::mfem::make_point_discretization (const discretization::backend::mfem::TermBinding< typename EquationT::Test, SchemaT > &residual_binding)

  Select the equation's physical point once; coordinate changes require a new selection.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  requires (discretization::domain::IsBoundary<equations::SupportOf<EquationT>> && discretization::blocks::SpatialTermTraits<typename EquationT::Test>::rank == 0)
  auto serif::operators::backend::mfem::make_boundary_discretization (const discretization::backend::mfem::TermBindin typename EquationT::Test, SchemaT > &residual_binding)

  Select the scalar equation's boundary through its declared schema support.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  auto serif::operators::backend::mfem::make_point_border_discretization (const discretization::backend::mfem::TermBi typename EquationT::Test, SchemaT > &residual_binding, const discretization::backend::mfem::TermBinding< typename EquationT::Border, SchemaT > &border_binding)

  Couple the selected scalar point to a global scalar on the same mesh.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  SelectedNodalValues serif::operators::backend::mfem::select_nodal_values (const NodalDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  Validate the whole supplied vector, but cache only coefficients selected by this equation.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  auto serif::operators::backend::mfem::make_nodal_dependency_context (const NodalDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

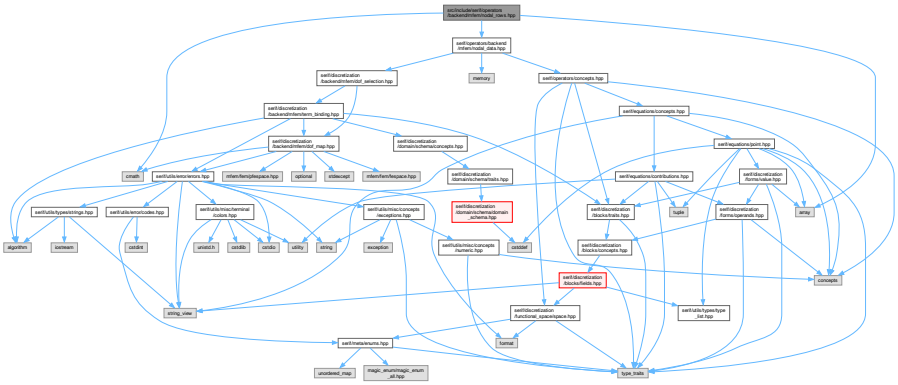
  The test term keys the selected coefficients needed by a nodal equation.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT>
  auto serif::operators::backend::mfem::make_point_border_dependency_context (const PointBorderDiscretization< EquationT, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  A border scalar is a separate tracked source; changing it does not reevaluate the point equation.

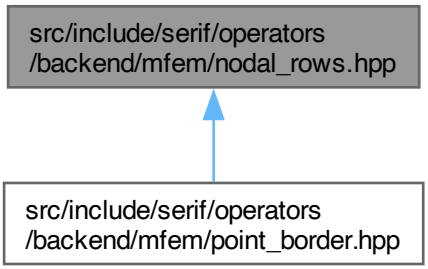
11.164 src/include/serif/operators/backend/mfem/nodal_rows.hpp File Reference

#include <array>
#include <cmath>

#include ”serif/operators/backend/mfem/nodal_data.hpp”
Include dependency graph for nodal_rows.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void serif::operators::backend::mfem::validate_nodal_output (const NodalData< EquationT, SchemaT > &data, const ::mfem::Vector &result)

  A frozen linearization still requires a current selection and a correctly sized output.

- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void serif::operators::backend::mfem::apply_residual_rows (const NodalData< EquationT, SchemaT > &data, ::mfem::Vector &result)

  Replace only selected entries of an assembled residual with the accepted nodal residual.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void serif::operators::backend::mfem::apply_jacobian_rows (const NodalData< EquationT, SchemaT > &data, const ::mfem::Vector &direction, ::mfem::Vector &result)

  Replace complete Jacobian rows, including every unconstrained coupling at the selected rows.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT> requires requires(const CoefficientsT& direction) { direction.template get<typename EquationT::Test>(); }
  void serif::operators::backend::mfem::apply_jacobian_rows (const NodalData< EquationT, SchemaT > &data, const CoefficientsT &direction, ::mfem::Vector &result)

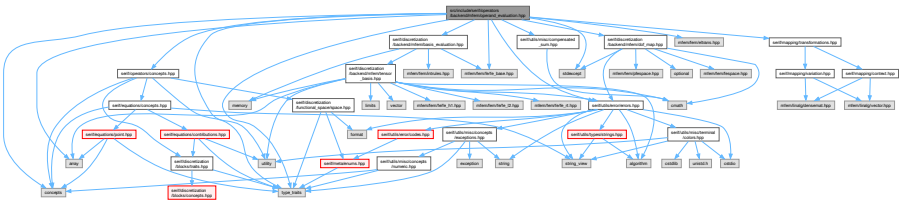
  Select the nodal direction by its physical term and verify its coefficient ordering.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void serif::operators::backend::mfem::mask_residual_dual (const NodalData< EquationT, SchemaT > &data, const ::mfem::Vector &dual, ::mfem::Vector &volume_dual)

  Remove constrained rows before applying the volume Jacobian transpose.
- template<NodalValueEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void serif::operators::backend::mfem::add_jacobian_transpose_rows (const NodalData< EquationT, SchemaT > &data, const ::mfem::Vector &dual, ::mfem::Vector &result)

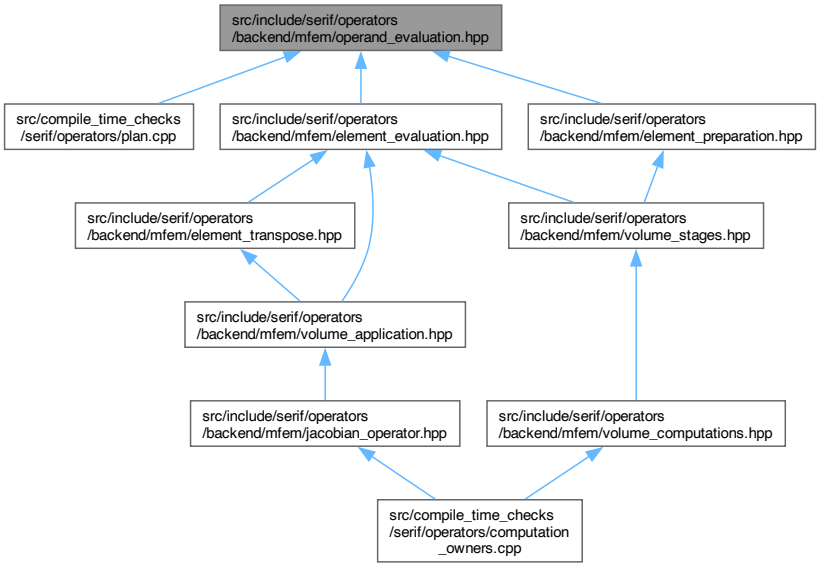
  Add the constrained-row transpose after applying the volume transpose to the masked dual.

11.165 src/include/serif/operators/backend/mfem/operand_↩ evaluation.hpp File Reference

#include <array>
#include <concepts>
#include <type_traits>
#include <stdexcept>
#include <utility>
#include "mfem/fem/fe_base.hpp"
#include "mfem/fem/eltrans.hpp"
#include "serif/operators/concepts.hpp"
#include "serif/discretization/backend/mfem/basis_evaluation.hpp"
#include "serif/discretization/backend/mfem/dof_map.hpp"
#include "serif/mapping/transformations.hpp"
#include "serif/utils/error/errors.hpp"
#include "serif/utils/misc/compensated_sum.hpp"
Include dependency graph for operand_evaluation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::OperandEvaluator< OperandT >

  Scalar values and H1/H(div) vector values or divergences share coefficient contractions.
- struct serif::operators::backend::mfem::BasisWorkspace

  Reusable scratch storage for basis contractions, reducing repeated allocations.
- class serif::operators::backend::mfem::TensorBasisCache

  Share immutable tensor plans within one operator; copied rules are keyed by their contents.

Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Concepts

- concept serif::operators::backend::mfem::EvaluableOperand

  Check the complete interface used by preparation, Jacobian application, and its transpose.

Enumerations

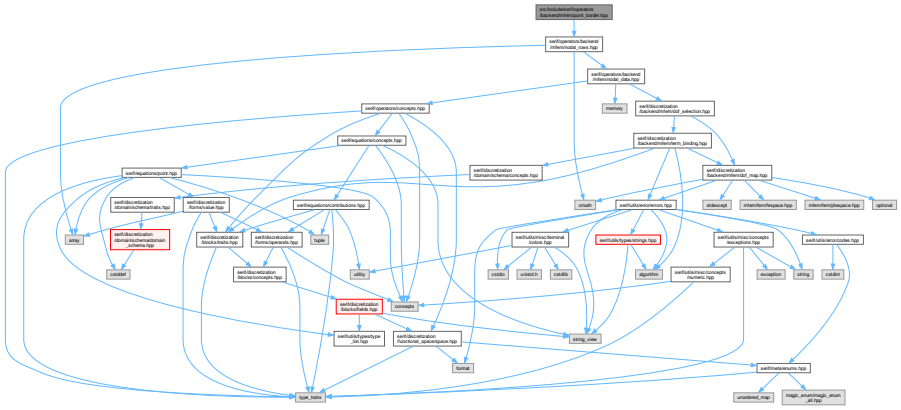
- enum class serif::operators::backend::mfem::ContractionMode

  Options enum to control how the tensor contractions are performed.
- enum class serif::operators::backend::mfem::ReferenceOperation

  Options enum to control what is actually evaluated at the reference points.

11.166 src/include/serif/operators/backend/mfem/point_border.hpp File Reference

#include "serif/operators/backend/mfem/nodal_rows.hpp"
Include dependency graph for point_border.hpp:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void serif::operators::backend::mfem::validate_point_border_outputs (const PointBorderData< EquationT, SchemaT > &data, const ::mfem::Vector &carrier, const ::mfem::Vector &phase)

  Point borders accept one carrier vector and one global scalar output.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT>
  void serif::operators::backend::mfem::add_point_border_residual (const PointBorderData< EquationT, SchemaT > &data, ::mfem::Vector &carrier, ::mfem::Vector &phase)

  Add the accepted border force to the carrier and set the scalar phase residual.

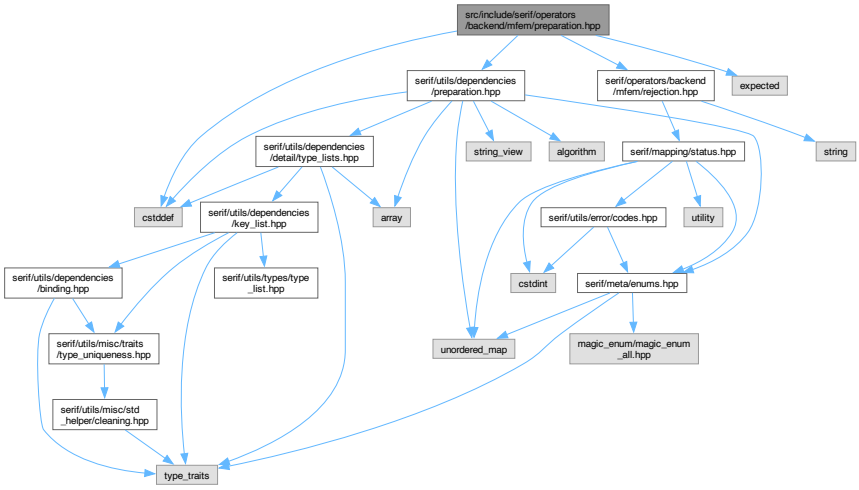
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT, typename CoefficientsT> void serif::operators::backend::mfem::apply_point_border_jacobian (const PointBorderData< EquationT, SchemaT > &data, const CoefficientsT &direction, ::mfem::Vector &carrier, ::mfem::Vector &phase)

  Apply the two off-diagonal Jacobian blocks of an affine scalar point border.
- template<AffinePointEquation EquationT, discretization::domain::schema::IsSchema SchemaT> void serif::operators::backend::mfem::apply_point_border_jacobian_transpose (const PointBorderData< EquationT, SchemaT > &data, const ::mfem::Vector &carrier_dual, const ::mfem::Vector &phase_↩ dual, ::mfem::Vector &carrier_result, ::mfem::Vector &border_result)

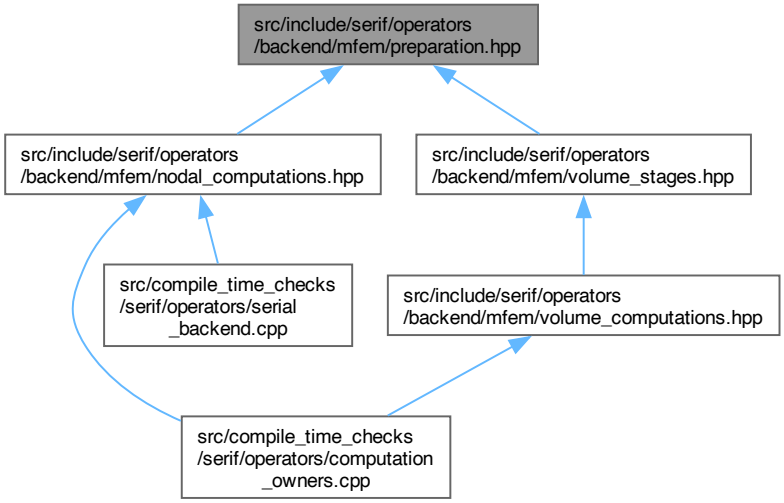
  Add the coordinate transpose of both border blocks to their coefficient duals.

11.167 src/include/serif/operators/backend/mfem/preparation.hpp File Reference

#include <cstddef>
#include <expected>
#include "serif/utils/dependencies/preparation.hpp"
#include "serif/operators/backend/mfem/rejection.hpp"
Include dependency graph for preparation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::NumericalWork

  Numerical work derived from the dependency engine's most recent report.

Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Typedefs

- using serif::operators::backend::mfem::PreparationMode = utils::dependencies::PreparationMode

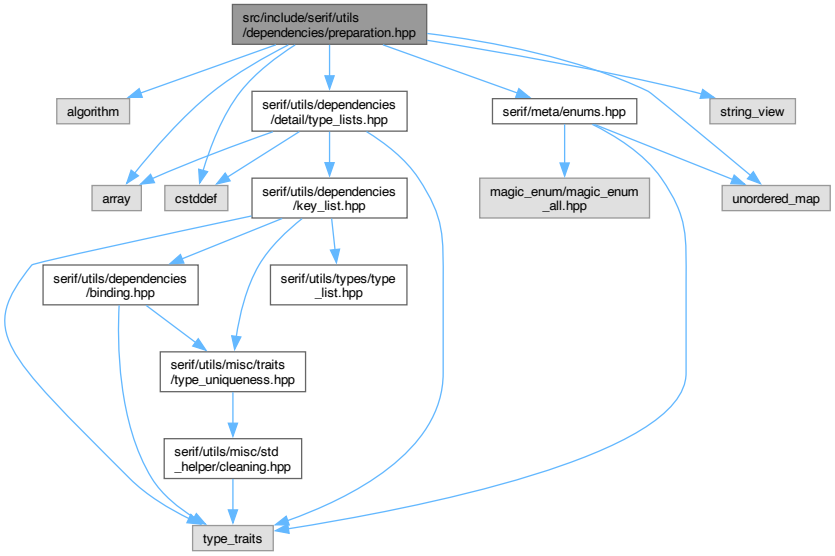
Functions

- template<typename CacheT, typename PrepareT>
  std::expected< void, PreparationRejection > serif::operators::backend::mfem::try_prepare_validated (CacheT &cache, PrepareT &&prepare)

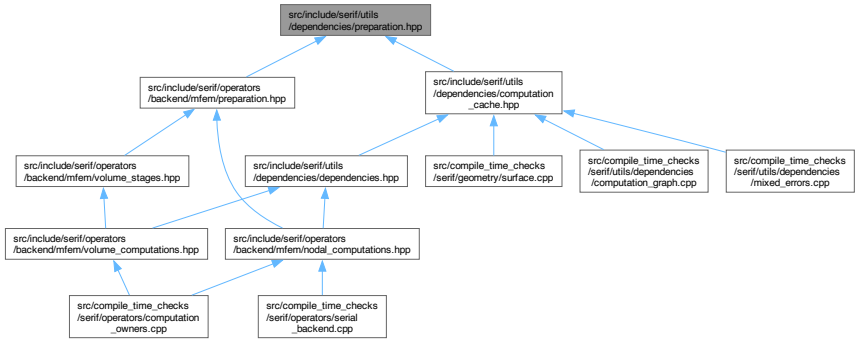
  A failed validation or calculation must make the previous accepted result inaccessible.

11.168 src/include/serif/utils/dependencies/preparation.hpp File Reference

#include <algorithm>
#include <array>
#include <cstddef>
#include <unordered_map>
#include <string_view>
#include "serif/utils/dependencies/detail/type_lists.hpp"
#include "serif/meta/enums.hpp"
Include dependency graph for preparation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::PreparationFailure< ErrorT >
- struct serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

Enumerations

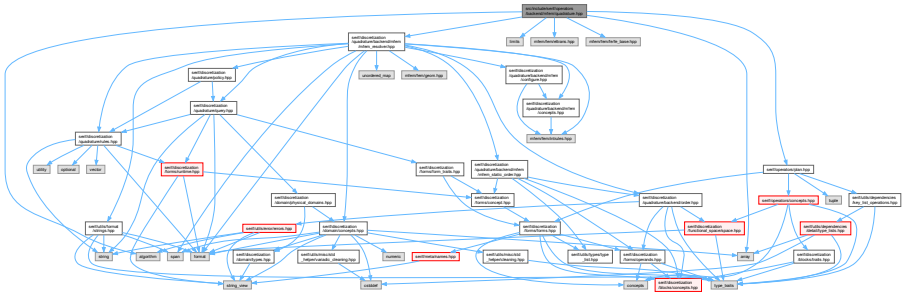
- enum class serif::utils::dependencies::PreparationMode { serif::utils::dependencies::reuse_unchanged , serif::utils::dependencies::rebuild }
- enum class serif::utils::dependencies::PreparationStatus { serif::utils::dependencies::not_required , serif::utils::dependencies::reused , serif::utils::dependencies::computed }

Variables

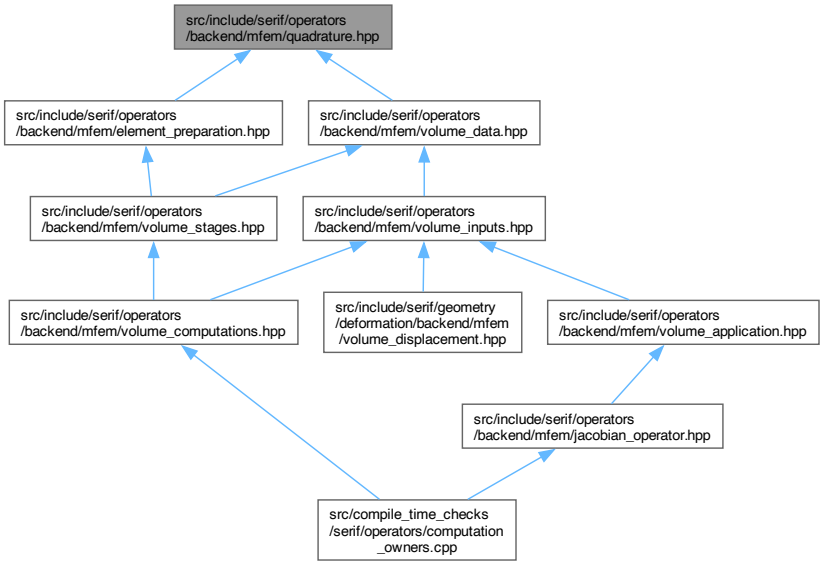
- std::unordered_map< PreparationStatus, std::string_view > serif::utils::dependencies::preparation_status_name_map = meta::enum_to_enum_string_map<PreparationStatus>()
- std::unordered_map< PreparationMode, std::string_view > serif::utils::dependencies::preparation_mode_name_map = meta::enum_to_enum_string_map<PreparationMode>()

11.169 src/include/serif/operators/backend/mfem/quadrature.hpp File Reference

#include <algorithm>
#include <array>
#include <limits>
#include "mfem/fem/eltrans.hpp"
#include "mfem/fem/fe/fe_base.hpp"
#include "serif/discretization/quadrature/backend/mfem/mfem_resolver.hpp"
#include "serif/operators/plan.hpp"
Include dependency graph for quadrature.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

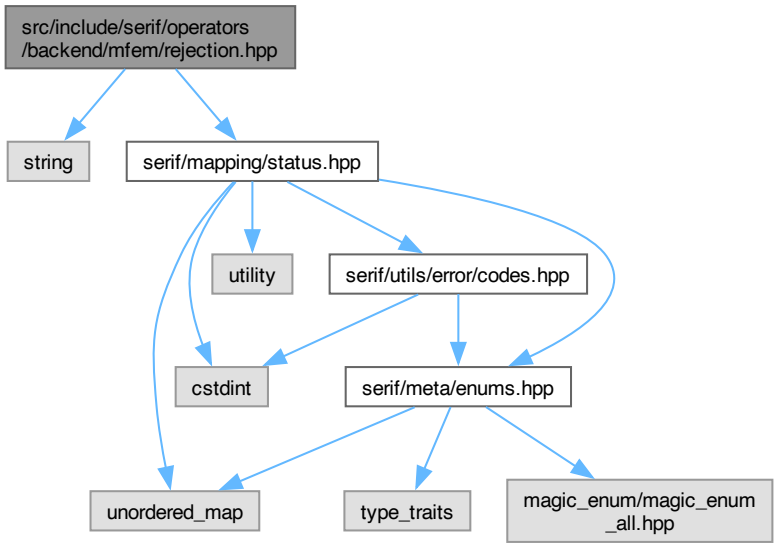
- template<VolumeEquation RowEquationT, typename ContributionT>
  const ::mfem::IntegrationRule & serif::operators::backend::mfem::equation_rule (const ContributionT &equation, const discretization::quadrature::Policy &policy, ::mfem::ElementTransformation &transformation, const ::mfem::FiniteElement &displacement_element, discretization::quadrature::MappingKind mapping_kind=discretization::quadrature::MappingKind::general)

  Resolve one rule covering all operand forms and the geometry weight.

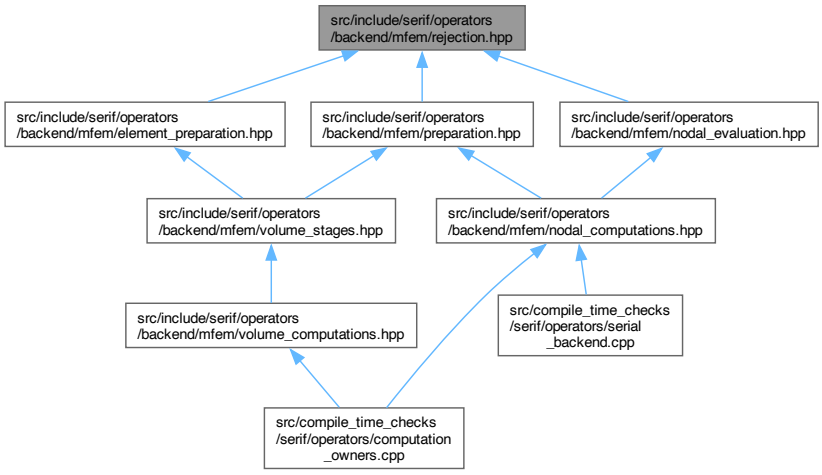
11.170 src/include/serif/operators/backend/mfem/rejection.hpp File Reference

#include <string>
#include "serif/mapping/status.hpp"

Include dependency graph for rejection.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::PreparationRejection

Namespaces

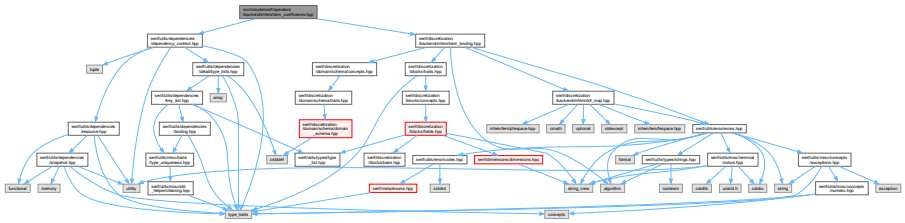
- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Enumerations

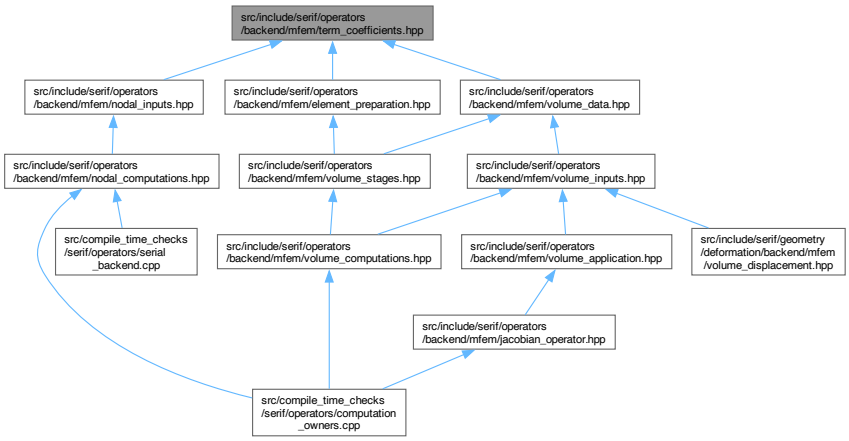
- enum class serif::operators::backend::mfem::PreparationRejectionReason { serif::operators::backend::mfem::MAPPING , serif::operators::backend::mfem::EQUATION , serif::operators::backend::mfem::NON_FINITE }

11.171 src/include/serif/operators/backend/mfem/term_coefficients.hpp File Reference

#include "serif/utils/dependencies/dependency_context.hpp"
#include "serif/discretization/backend/mfem/term_binding.hpp"
Include dependency graph for term_coefficients.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::MappingCoefficients
- struct serif::operators::backend::mfem::CoefficientView< BindingT >

Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

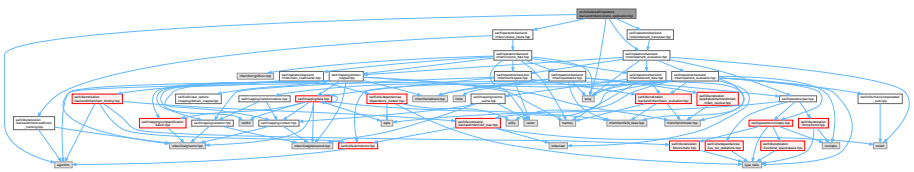
- template<typename BindingT>
  CoefficientView< BindingT > serif::operators::backend::mfem::bind (const BindingT &binding, const ::mfem::Vector &values)

  Borrow a term binding and its coefficients without extending their lifetimes.
- template<typename BindingT>
  void serif::operators::backend::mfem::bind (const BindingT &, ::mfem::Vector &&)=delete
- template<typename BindingT>
  void serif::operators::backend::mfem::bind (const BindingT &, const ::mfem::Vector &&)=delete
- template<typename BindingT>
  void serif::operators::backend::mfem::bind (const BindingT &&, const ::mfem::Vector &)=delete
- template<typename... BindingTs>
  auto serif::operators::backend::mfem::make_term_coefficients (BindingTs... bindings)

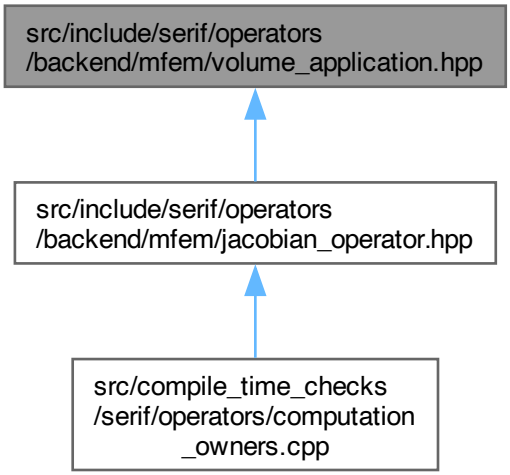
  Associate each physical term with its bound coefficients in an ordinary dependency context.

11.172 src/include/serif/operators/backend/mfem/volume_↩ application.hpp File Reference

#include <algorithm>
#include <array>
#include "serif/operators/backend/mfem/volume_inputs.hpp"
#include "serif/operators/backend/mfem/element_evaluation.hpp"
#include "serif/operators/backend/mfem/element_transpose.hpp"
Include dependency graph for volume_application.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

- template<VolumeEquation EquationT, typename SchemaT>
  void serif::operators::backend::mfem::validate_volume_data (const VolumeData< EquationT, SchemaT > &data)

  Reject changes to discretization objects borrowed by an accepted linearization.
- template<VolumeEquation EquationT, typename SchemaT, typename VisitT>
  void serif::operators::backend::mfem::for_each_volume_element (const VolumeData< EquationT, SchemaT > &data, VisitT &&visit)

  Visit each contribution's own batch in equation order, without scanning other contributions.
- template<VolumeEquation EquationT, typename SchemaT>
  int serif::operators::backend::mfem::volume_height (const VolumeData< EquationT, SchemaT > &data)

  Return the supported residual size.
- template<VolumeEquation EquationT, typename SchemaT>
  int serif::operators::backend::mfem::volume_width (const VolumeData< EquationT, SchemaT > &data)

  Packed directions contain unique coefficient terms followed by displacement.

- template<VolumeEquation EquationT, typename SchemaT>
  std::size_t serif::operators::backend::mfem::tensor_element_count (const VolumeData< EquationT, SchemaT > &data)

  Count contribution-elements using tensor contractions for any sampled basis.
- template<VolumeEquation EquationT, typename SchemaT>
  int serif::operators::backend::mfem::integration_order (const VolumeData< EquationT, SchemaT > &data, int element)

  Return the largest contribution quadrature order on a supported mesh element.
- template<VolumeEquation EquationT, typename SchemaT>
  void serif::operators::backend::mfem::volume_residual (const VolumeData< EquationT, SchemaT > &data, ::mfem::Vector &result)

  Copy the residual already assembled and checked when this result was accepted.
- template<typename RowBindingT>
  void serif::operators::backend::mfem::accumulate_element_result (const RowBindingT &residual_↩ binding, int element, Workspace &workspace, utils::CompensatedSum &global_sum)

  Global rows accumulate with compensation; spatial rows scatter into their supported DOFs.
- template<VolumeEquation EquationT, typename SchemaT>
  void serif::operators::backend::mfem::apply_volume_direction (const VolumeData< EquationT, SchemaT > &data, const std::array< const ::mfem::Vector *, EquationPlan< EquationT >::column↩ _count > &directions, const ::mfem::Vector &displacement_direction, Workspace &workspace, ::mfem::Vector &result)

  Apply the element linearizations and assemble a supported residual direction.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  void serif::operators::backend::mfem::apply_jacobian (const VolumeData< EquationT, SchemaT > &data, const CoefficientsT &direction, const ::mfem::Vector &displacement_direction, Workspace &workspace, ::mfem::Vector &result)

  Apply a direction whose terms explicitly name the accepted coefficient bindings.
- template<VolumeEquation EquationT, typename SchemaT>
  void serif::operators::backend::mfem::apply_jacobian (const VolumeData< EquationT, SchemaT > &data, const ::mfem::Vector &direction, Workspace &workspace, ::mfem::Vector &result)

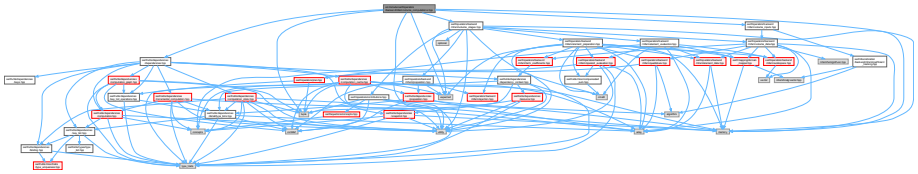
  Unpack unique coefficient terms and displacement, then use the same numerical action.
- template<VolumeEquation EquationT, typename SchemaT>
  void serif::operators::backend::mfem::apply_jacobian_transpose (const VolumeData< EquationT, SchemaT > &data, const ::mfem::Vector &residual_dual, Workspace &workspace, ::mfem::Vector &result)

  Assemble the full coordinate transpose in the same packed order as the forward action.

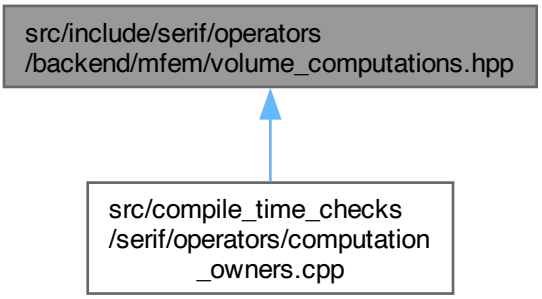
11.173 src/include/serif/operators/backend/mfem/volume_↩ computations.hpp File Reference

```
#include <expected>
#include <memory>
#include <tuple>
#include <utility>
#include "serif/operators/backend/mfem/volume_stages.hpp"
#include "serif/operators/backend/mfem/volume_inputs.hpp"
```

#include "serif/utils/dependencies/dependencies.hpp"
Include dependency graph for volume_computations.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Typedefs

- template<std::size_t Contribution, typename OperandT>
  using serif::operators::backend::mfem::VolumeOperandResult = std::conditional_t<OperandEvaluator<OperandT>↩
  ::geometry_dependent, volume_keys::PhysicalValues<Contribution, OperandT>, volume_keys::ReferenceValues<Cont
  OperandT>>

  Values need a mapping node only when the operand's physical representation depends on geometry.
- template<typename CacheT>
  using serif::operators::backend::mfem::VolumeEquationOf = typename std::remove_cvref_t<decltype(std↩
  ::declval<const CacheT&>().template get<volume_keys::Linearization>())>::Equation

  Recover the equation type from a cache's accepted result type, without accessing a result.

Functions

- template<std::size_t Contribution, typename OperandT, VolumeEquation EquationT, typename SchemaT>
  auto serif::operators::backend::mfem::make_operand_computations (const std::shared_ptr< VolumeScratch > &scratch)

  Declare one coefficient contraction, followed by a physical mapping when the space requires it.
- template<std::size_t Index, VolumeEquation EquationT, typename SchemaT, typename ContributionT, typename... Operands>
  auto serif::operators::backend::mfem::make_contribution_computations (const std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > &configuration, const std::shared_ptr< VolumeScratch > &scratch, ContributionBinding< ContributionT > contribution, utils::types::TypeList< Operands... >)

  Build one contribution's branch in the same order as its numerical description.
- template<VolumeEquation EquationT, typename SchemaT, std::size_t... Indices>
  auto serif::operators::backend::mfem::connect_volume_contributions (const std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > &configuration, std::index_sequence< Indices... >)

  Expose contribution computations so deformation and equations can share one dependency graph.
- template<VolumeEquation EquationT, std::size_t Index, typename CacheT>
  void serif::operators::backend::mfem::report_contribution_preparation (const CacheT &cache, NumericalWork &report)

  Convert the dependency engine's node statuses to numerical work counts for one contribution.
- template<VolumeEquation EquationT, typename SchemaT>
  auto serif::operators::backend::mfem::make_volume_equation_computations (EquationT equation, const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, VolumeOptions options={})

  Declare reusable volume computations without selecting their source of displacement.
- template<VolumeEquation EquationT, typename SchemaT, typename InputsT>
  auto serif::operators::backend::mfem::make_volume_equation_cache (EquationT equation, const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const InputsT &inputs, VolumeOptions options={})

  Construct a standalone cache from the same computations used by a coupled deformation graph.
- template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
  std::expected< void, PreparationRejection > serif::operators::backend::mfem::try_prepare_volume (CacheT &cache, const DiscretizationT &discretization, const CoefficientsT &term_coefficients, const ::mfem::Vector &displacement_coefficients, PreparationMode mode=PreparationMode::reuse_unchanged)

  Validate current inputs and atomically publish a complete volume linearization.
- template<typename CacheT, typename DiscretizationT, typename CoefficientsT>
  void serif::operators::backend::mfem::prepare_volume (CacheT &cache, const DiscretizationT &discretization, const CoefficientsT &term_coefficients, const ::mfem::Vector &displacement_coefficients, PreparationMode mode=PreparationMode::reuse_unchanged)

  Throw a caller-visible numerical error when a complete preparation cannot be accepted.
- template<typename CacheT>
  NumericalWork serif::operators::backend::mfem::volume_work_report (const CacheT &cache)

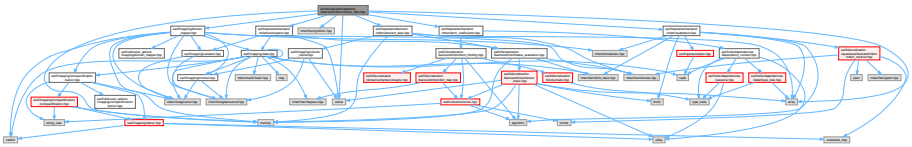
  Derive work counts from accepted node statuses; no second report lifecycle is stored.

11.174 src/include/serif/operators/backend/mfem/volume_data.hpp File Reference

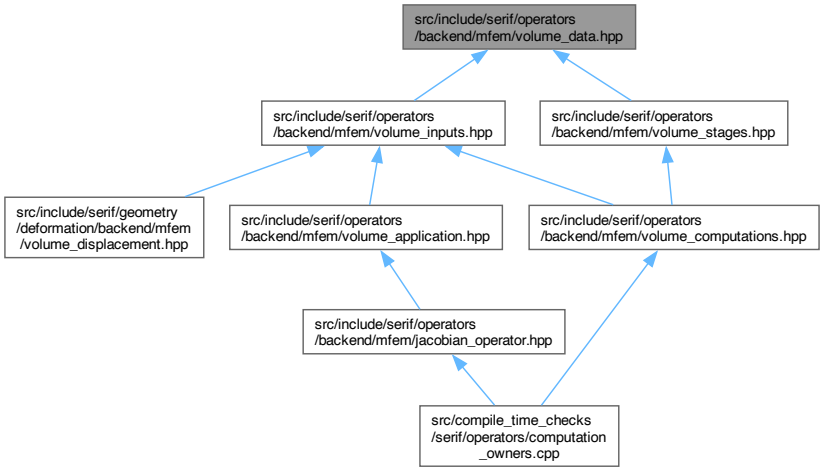
#include <array>
#include <memory>

```
#include <vector>
#include "mfem/fem/gridfunc.hpp"
#include "serif/mapping/domain_mapper.hpp"
#include "serif/operators/backend/mfem/term_coefficients.hpp"
#include "serif/operators/backend/mfem/element_data.hpp"
#include "serif/operators/backend/mfem/quadrature.hpp"
#include "serif/operators/backend/mfem/workspace.hpp"
```

Include dependency graph for volume_data.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::volume_keys::Basis< Contribution >
- struct serif::operators::backend::mfem::volume_keys::Geometry< Contribution >
- struct serif::operators::backend::mfem::volume_keys::ReferenceValues< Contribution, OperandT >
- struct serif::operators::backend::mfem::volume_keys::PhysicalValues< Contribution, OperandT >
- struct serif::operators::backend::mfem::volume_keys::QuadratureLinearization< Contribution >
- struct serif::operators::backend::mfem::volume_keys::Linearization
- struct serif::operators::backend::mfem::BasisBindings< ColumnCount >
- struct serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >

  Borrowed discretization objects must outlive their graph and its accepted results.
- struct serif::operators::backend::mfem::VolumeOptions

  Numerical choices fixed when the graph is constructed.

- struct serif::operators::backend::mfem::OperandSamples

  Reference samples remain available for derivatives through the physical mapping.
- struct serif::operators::backend::mfem::ContributionData< OperandCount >
- struct serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >

  Stable equation ownership also retains temporary and move-only contribution trees.
- struct serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >

  One accepted result contains ordered contribution batches and the assembled residual.
- struct serif::operators::backend::mfem::VolumeScratch

  Scratch belongs to graph callables and is never a dependency source.

Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem
- namespace serif::operators::backend::mfem::volume_keys

Typedefs

- template<typename ElementT>
  using serif::operators::backend::mfem::ElementBatch = std::vector<std::shared_ptr<const ElementT>>

  A stage owns immutable element results, shared with downstream stages and accepted linearizations.

Enumerations

- enum class serif::operators::backend::mfem::ContractionMode

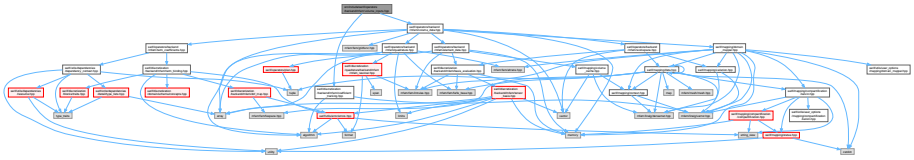
  Options enum to control how the tensor contractions are performed.

Functions

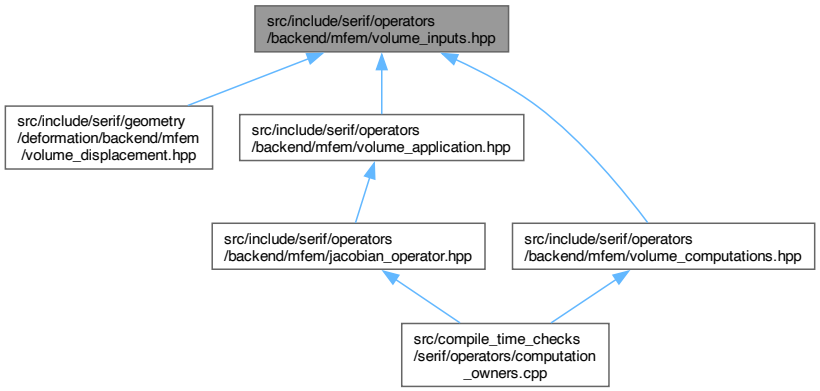
- template<typename TestT, typename SchemaT>
  serif::operators::backend::mfem::VolumeDiscretization (const discretization::backend::mfem::TermBinding< TestT, SchemaT > &, const discretization::backend::mfem::TermBinding< discretization::blocks::displacement::vector, SchemaT > &, const ::mfem::GridFunction &, const mapping::DomainMapper &) -> VolumeDiscretization< TestT, SchemaT >

11.175 src/include/serif/operators/backend/mfem/volume_inputs.hpp File Reference

#include "serif/operators/backend/mfem/volume_data.hpp"
#include "serif/discretization/backend/mfem/coefficient_tracking.hpp"
Include dependency graph for volume_inputs.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::MetadataTracking

  Metadata is a value; moving its temporary wrapper does not change the mesh or bindings it describes.
- struct serif::operators::backend::mfem::MeshAndCoefficientBindings
- struct serif::operators::backend::mfem::DisplacementCoefficients
- struct serif::operators::backend::mfem::CompactificationCoefficients
- struct serif::operators::backend::mfem::CoefficientTracking

  Track coefficient values using an owned snapshot and the identity of their binding.

Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Functions

- template<typename TestT, typename SchemaT>
  void serif::operators::backend::mfem::validate_volume_discretization (const VolumeDiscretization< TestT, SchemaT > &discretization)

  Check borrowed mesh objects before sampling or applying accepted results.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT, typename... Columns>
  auto serif::operators::backend::mfem::make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const CoefficientsT &term_coefficients, utils::types::TypeList< Columns... >)

  Supply precisely the consumed terms; unrelated coefficients are not graph dependencies.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  auto serif::operators::backend::mfem::make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const CoefficientsT &term_coefficients)

  Bind equation and reference inputs when displacement will be produced inside the dependency graph.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT, typename... Columns>
  auto serif::operators::backend::mfem::make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const CoefficientsT &term_coefficients, const ::mfem::Vector &displacement_coefficients, utils::types::TypeList< Columns... > columns)

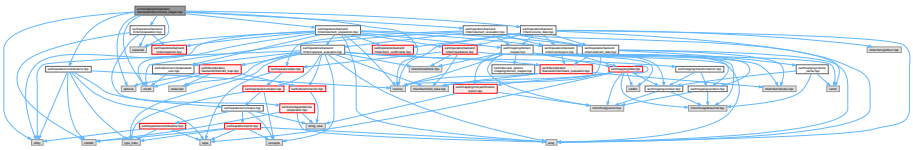
  Append a checked external displacement while preserving each original resource's tracking policy.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  auto serif::operators::backend::mfem::make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &discretization, const CoefficientsT &term_coefficients, const ::mfem::Vector &displacement_coefficients)

  Build a checked source context without retaining the coefficient vectors.
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  void serif::operators::backend::mfem::make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &, const CoefficientsT &, ::mfem::Vector &&)=delete
- template<VolumeEquation EquationT, typename SchemaT, typename CoefficientsT>
  void serif::operators::backend::mfem::make_volume_inputs (const VolumeDiscretization< typename EquationT::Test, SchemaT > &, const CoefficientsT &, const ::mfem::Vector &&)=delete

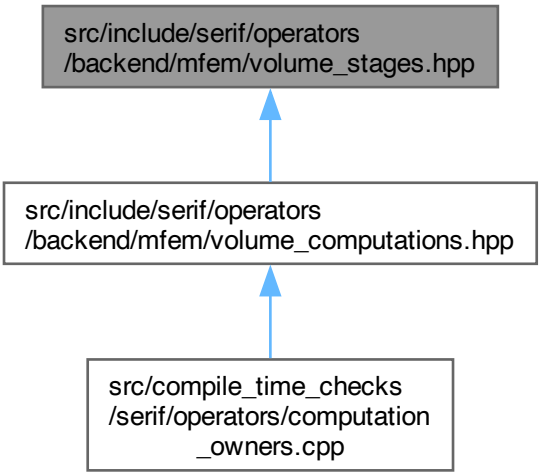
11.176 src/include/serif/operators/backend/mfem/volume_stages.hpp File Reference

#include <cmath>
#include <expected>
#include <memory>
#include <optional>
#include <utility>
#include "serif/operators/contributions.hpp"
#include "serif/operators/backend/mfem/element_preparation.hpp"
#include "serif/operators/backend/mfem/element_evaluation.hpp"
#include "serif/operators/backend/mfem/preparation.hpp"

#include ”serif/operators/backend/mfem/volume_data.hpp”
Include dependency graph for volume_stages.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

Typedefs

- template<typename OperandT>
  using serif::operators::backend::mfem::SamplesFor = OperandSamples

  Associate an operand type pack with equally ordered sample arguments.

Functions

- template<VolumeEquation EquationT, typename SchemaT, typename ContributionT>
auto serif::operators::backend::mfem::sample_contribution_basis (const VolumeConfiguration< EquationT, SchemaT > &configuration, const ContributionBinding< ContributionT > &contribution, const BasisBindings< EquationPlan< EquationT >::column_count > &bindings, TensorBasisCache &cache)

  Sample the reference discretization once for each supported element of this contribution.
- template<VolumeEquation EquationT, typename OperandT, typename BindingT>
OperandSamples serif::operators::backend::mfem::sample_contribution_operand (const ElementBatch< ElementBasis< EquationPlan< EquationT >::operand_count > > &basis, const CoefficientView< BindingT > &input, Workspace &workspace)

  Only this operand's coefficients and reference basis determine its reference samples.
- template<VolumeEquation EquationT, typename SchemaT, typename ContributionT>
std::expected< ElementBatch< ElementGeometry< EquationPlan< EquationT >::operand_count > >, PreparationRejection > serif::operators::backend::mfem::map_contribution_geometry (const VolumeConfiguration< EquationT, SchemaT > &configuration, const ElementBatch< ElementBasis< EquationPlan< EquationT >::operand_count > > &basis, const CoefficientView< typename VolumeConfiguration< EquationT, SchemaT >::DisplacementBinding > &displacement, const MappingCoefficients &compactification, Workspace &workspace)

  Freeze mapping data and mapped bases independently of the equation's coefficient inputs.
- template<typename OperandT, std::size_t OperandCount>
OperandSamples serif::operators::backend::mfem::map_contribution_operand (const OperandSamples &samples, const ElementBatch< ElementGeometry< OperandCount > > &geometry, Workspace &workspace)

  Map only operands whose physical representation depends on deformation.
- template<VolumeEquation EquationT, typename ContributionT, typename... Operands>
std::expected< ContributionData< EquationPlan< EquationT >::operand_count >, PreparationRejection > serif::operators::backend::mfem::evaluate_contribution_equation (const ContributionBinding< ContributionT > &contribution, const ElementBatch< ElementBasis< EquationPlan< EquationT >::operand_count > > &basis, const ElementBatch< ElementGeometry< EquationPlan< EquationT >::operand_count > > &geometry, Workspace &workspace, const SamplesFor< Operands > &... samples)

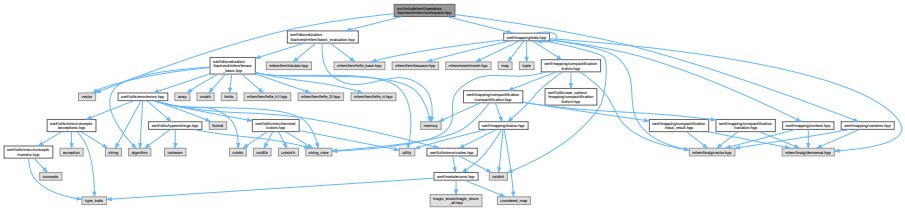
  Evaluate one equation contribution; its basis, mapping and individual operand samples are already prepared.
- template<VolumeEquation EquationT, typename SchemaT, typename... Contributions>
std::expected< VolumeData< EquationT, SchemaT >, PreparationRejection > serif::operators::backend::mfem::collect_
(const std::shared_ptr< const VolumeConfiguration< EquationT, SchemaT > > &configuration, const BasisBindings< EquationPlan< EquationT >::column_count > &bindings, Workspace &workspace, const Contributions &... contributions)

  Assemble once, retaining contribution order and compensated global scalar accumulation.

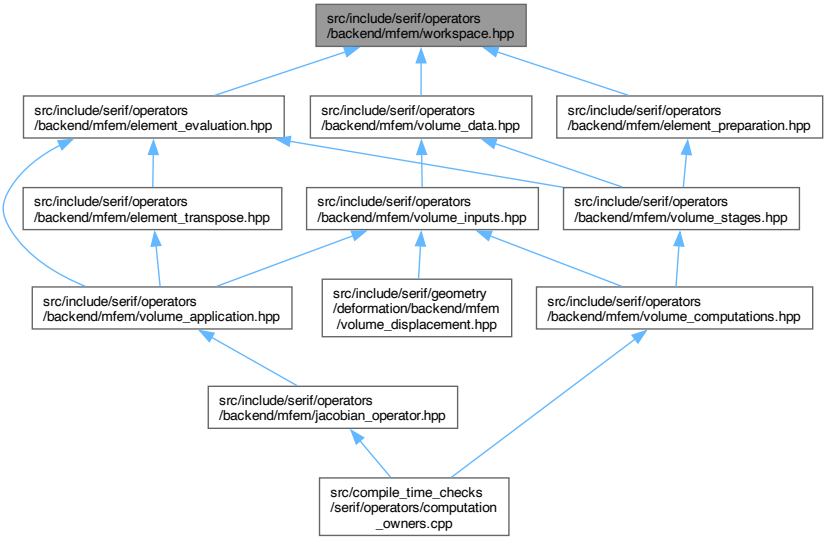
11.177 src/include/serif/operators/backend/mfem/workspace.hpp File Reference

#include <vector>
#include "serif/mapping/data.hpp"
#include "serif/mapping/variation.hpp"

#include ”serif/discretization/backend/mfem/basis_evaluation.hpp”
Include dependency graph for workspace.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::backend::mfem::Workspace

  Reusable scratch for applying the accepted element linearization.

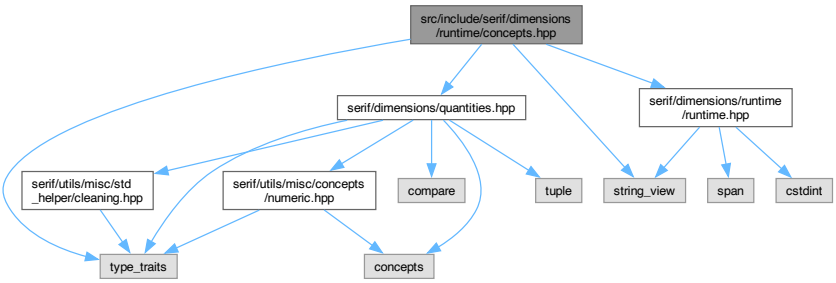
Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::backend
- namespace serif::operators::backend::mfem

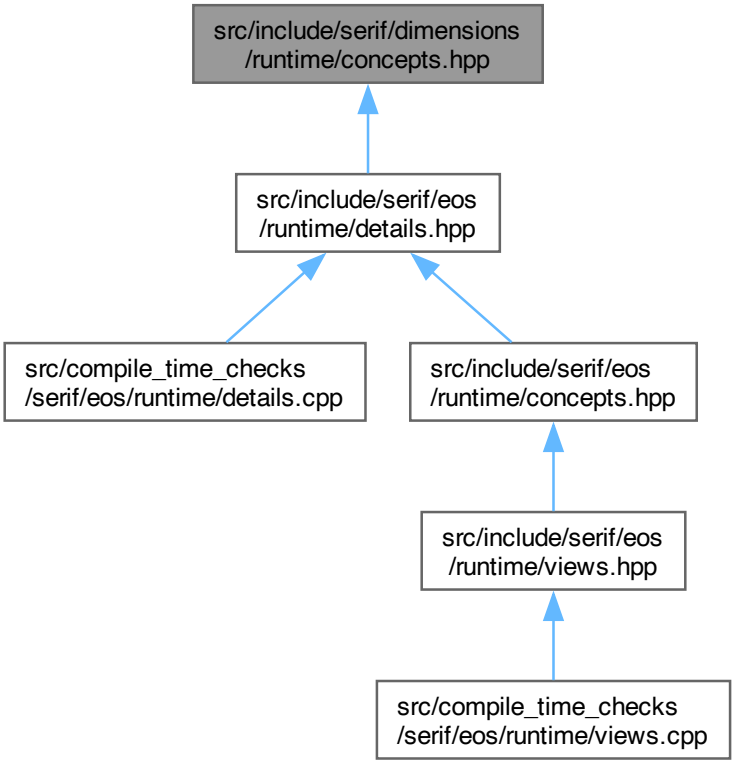
11.178 src/include/serif/dimensions/runtime/concepts.hpp File Reference

Runtime concept for a type that can be projected to a runtime type. This is useful when converting from the compile time type system to something that can be inspected at runtime.

#include <string_view>
#include <type_traits>
#include "serif/dimensions/quantities.hpp"
#include "serif/dimensions/runtime/runtime.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::dimensions
- namespace serif::dimensions::runtime

Concepts

- concept serif::dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity

Variables

- template<RuntimeIdentifiedThermodynamicQuantity Quantity>
  constexpr ThermodynamicQuantityID serif::dimensions::runtime::thermodynamicQuantityID {std↩
  ::string_view{Quantity::identifier}}

  Extract a compile time identifier from a thermodynamic quantity and project it into a runtime string_view.

11.178.1 Detailed Description

Runtime concept for a type that can be projected to a runtime type. This is useful when converting from the compile time type system to something that can be inspected at runtime.

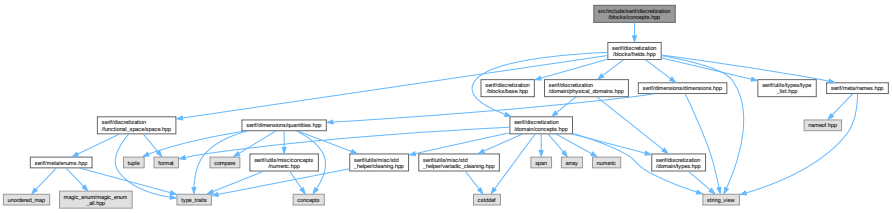
Author

Emily M. Boudreaux

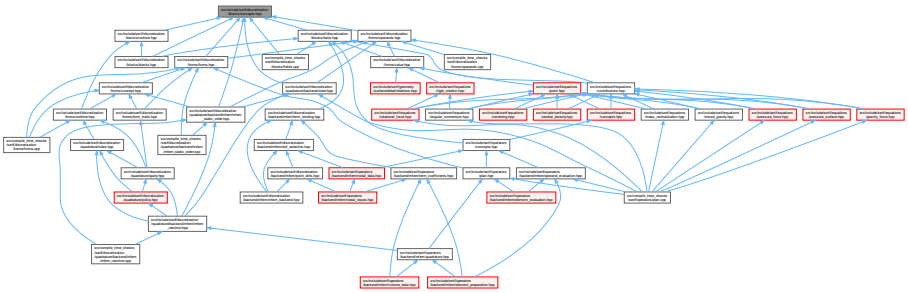
11.179 src/include/serif/discretization/blocks/concepts.hpp File Reference

This file defines concepts for blocks in the discretization framework.

#include "serif/discretization/blocks/fields.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::blocks::IsSpatialTerm< T >

  Helper struct to determine if a type is a spatial term (SpatialTerm).
- struct serif::discretization::blocks::IsGlobalScalarTerm< T >

  Helper struct to determine if a type is a global scalar term (GlobalScalarTerm).
- struct serif::discretization::blocks::IsPhysicalSpatialTerm< T >

  Helper struct to determine if a type is a physical spatial term (PhysicalSpatialTerm).

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::blocks

Concepts

- concept serif::discretization::blocks::Field

  Concept for field types in the discretization framework. To be a field a type must derive from the field base class.
- concept serif::discretization::blocks::Term

  Concept for term types in the discretization framework. To be a term a type must derive from term_base, must have a field_type member, and that field_type member must be a valid field.
- concept serif::discretization::blocks::HasFunctionalSpace

  Concept for term types that have a functional space. To have a functional space a type must first be a valid Term and must have a FunctionalSpace member type.
- concept serif::discretization::blocks::HasSupport

  Concept for term types that have a domain over which they are supported. To have support a type must first be a valid Term and must have a Support member type.
- concept serif::discretization::blocks::HasQuantity

  Concept for term types that have a physical quantity associated with them. To have a quantity a type must first be a valid Term and must have a Quantity member type.
- concept serif::discretization::blocks::HasValueBlock

  Concept for term types that have a value block associated with them. To have a value block a type must first be a valid Term and must have a value member type.
- concept serif::discretization::blocks::HasResidualBlock

  Concept for term types that have a residual block associated with them. To have a residual block a type must first be a valid Term and must have a residual member type.
- concept serif::discretization::blocks::SpatialTerm

  Concept for term types that are spatial terms. A spatial term is a term that has a functional space (HasFunctionalSpace) and a support (HasSupport).
- concept serif::discretization::blocks::GlobalScalarTerm

  Concept for term types that are global scalar terms. A global scalar term is a term that has support (HasSupport) but does not have a functional space (HasFunctionalSpace) and is derived from the global_scalar class.
- concept serif::discretization::blocks::PhysicalSpatialTerm

  Concept for a physical spatial term. A physical spatial term is a spatial term (SpatialTerm) that has a physical quantity associated with it (HasQuantity).

11.179.1 Detailed Description

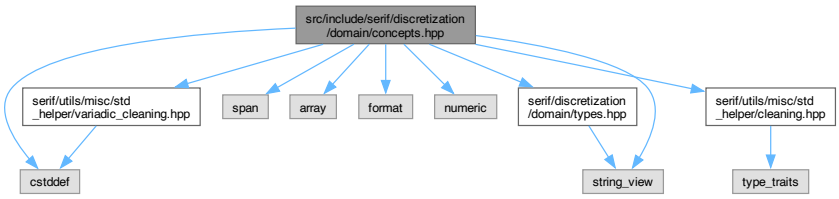
This file defines concepts for blocks in the discretization framework.

Author

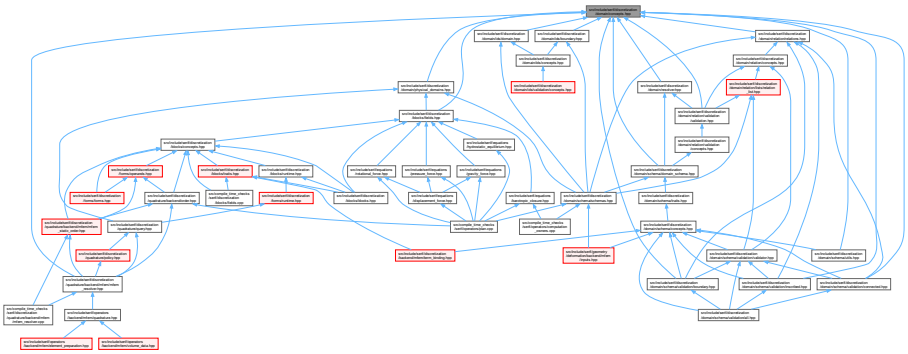
Emily M. Boudreaux

11.180 src/include/serif/discretization/domain/concepts.hpp File Reference

#include <cstddef>
#include <string_view>
#include
#include <array>
#include <format>
#include <numeric>
#include "serif/discretization/domain/types.hpp"
#include "serif/utils/misc/std_helper/cleaning.hpp"
#include "serif/utils/misc/std_helper/variadic_cleaning.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::discretization::domain::DomainSet< DomainTs >
- class serif::discretization::domain::DomainOrSet
- struct serif::discretization::domain::BoundarySet< BoundaryTs >
- struct serif::discretization::domain::DomainOrSetList< DomainTs >
- struct std::formatter< serif::discretization::domain::DomainOrSet >

Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain

Concepts

- concept serif::discretization::domain::IsDomain
- concept serif::discretization::domain::IsBoundary
- concept serif::discretization::domain::IsDomainSet
- concept serif::discretization::domain::IsDomainOrSet
- concept serif::discretization::domain::DomainIsIdentifiedBy
- concept serif::discretization::domain::BoundaryIsIdentifiedBy

Functions

- template<typename... Ts>
  consteval std::size_t serif::discretization::domain::CountVariadicArguments () noexcept

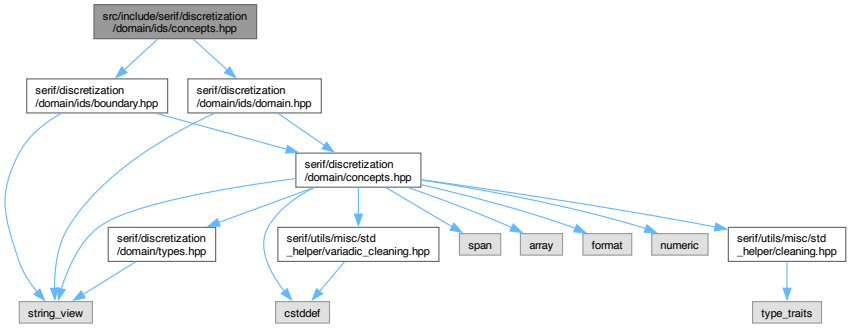
Variables

- template<typename T>
  constexpr bool serif::discretization::domain::is_domain_set_v = false
- template<IsDomain... DomainTs>
  constexpr bool serif::discretization::domain::is_domain_set_v< DomainSet< DomainTs... > > = true
- template<class Child, class Parent>
  constexpr bool serif::discretization::domain::AIsBaseClassOfB

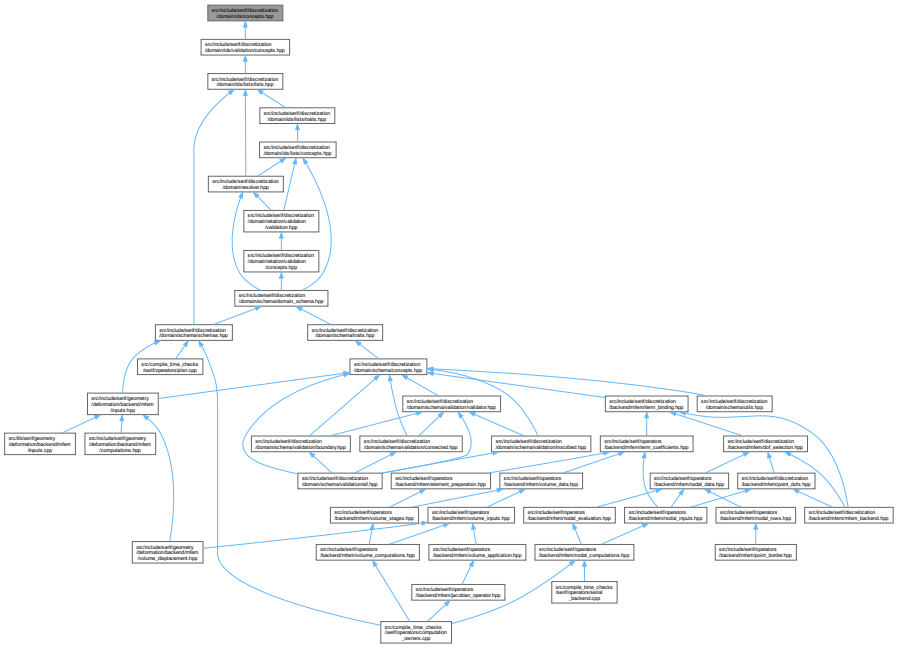
11.181 src/include/serif/discretization/domain/ids/concepts.hpp File Reference

This file defines concepts for domain and boundary IDs in the discretization framework. These are specifically used to validate that a type is a valid DomainID or BoundaryID.

#include "serif/discretization/domain/ids/boundary.hpp"
#include "serif/discretization/domain/ids/domain.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::ids

Concepts

- concept serif::discretization::domain::ids::IsDomainID

  Concept for types that are domain IDs in the discretization framework. To be a domain ID a type must satisfy the is_domain_id_v trait.
- concept serif::discretization::domain::ids::IsBoundaryID

  Concept for types that are boundary IDs in the discretization framework. To be a boundary ID a type must satisfy the is_boundary_id_v trait.
- concept serif::discretization::domain::ids::IsDomainOrBoundaryID

  Concept for types that are either domain IDs or boundary IDs in the discretization framework.

Variables

- template<typename T>
  constexpr bool serif::discretization::domain::ids::is_domain_id_v = false

  Trait to determine if a type is a domain ID. Defaults to false for all types.

- template<IsDomain DomainT, int IdentifierValue>
  constexpr bool serif::discretization::domain::ids::is_domain_id_v< DomainID< DomainT, IdentifierValue > > = true

  Trait to determine if a type is a domain ID. Specialization which is true for valid DomainID types.
- template<typename T>
  constexpr bool serif::discretization::domain::ids::is_boundary_id_v = false

  Trait to determine if a type is a boundary ID. Defaults to false for all types.
- template<IsBoundary BoundaryT, int IdentifierValue>
  constexpr bool serif::discretization::domain::ids::is_boundary_id_v< BoundaryID< BoundaryT, IdentifierValue > > = true

  Trait to determine if a type is a boundary ID. Specialization which is true for valid BoundaryID types.

11.181.1 Detailed Description

This file defines concepts for domain and boundary IDs in the discretization framework. These are specifically used to validate that a type is a valid DomainID or BoundaryID.

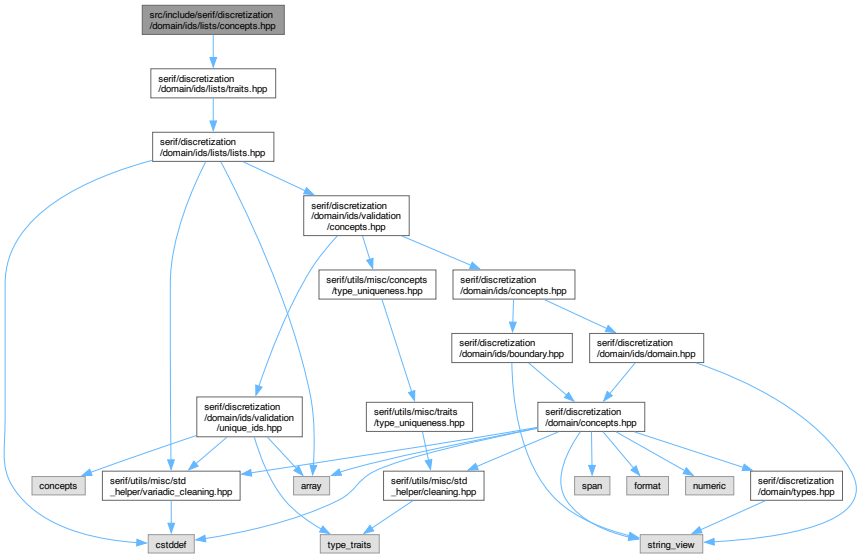
Author

Emily M. Boudreaux

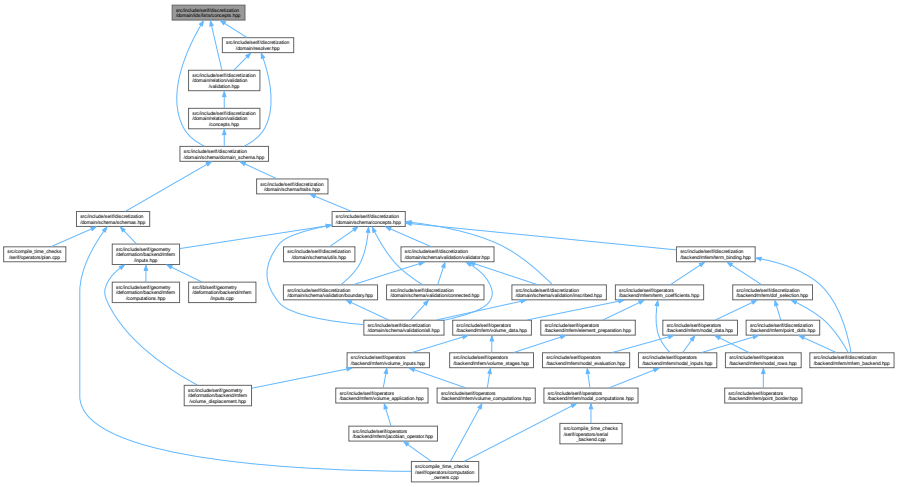
11.182 src/include/serif/discretization/domain/ids/lists/concepts.hpp File Reference

This file defines concepts for domain ID lists in the discretization framework.

#include "serif/discretization/domain/ids/lists/traits.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::ids
- namespace serif::discretization::domain::ids::lists

Concepts

- concept serif::discretization::domain::ids::lists::IsDomainIDList

  Concept for types that are domain ID lists in the discretization framework. To be a domain ID list a type must satisfy the is_domain_id_list_v trait.
- concept serif::discretization::domain::ids::lists::IsBoundaryIDList

  Concept for types that are boundary ID lists in the discretization framework. To be a boundary ID list a type must satisfy the is_boundary_id_list_v trait.

11.182.1 Detailed Description

This file defines concepts for domain ID lists in the discretization framework.

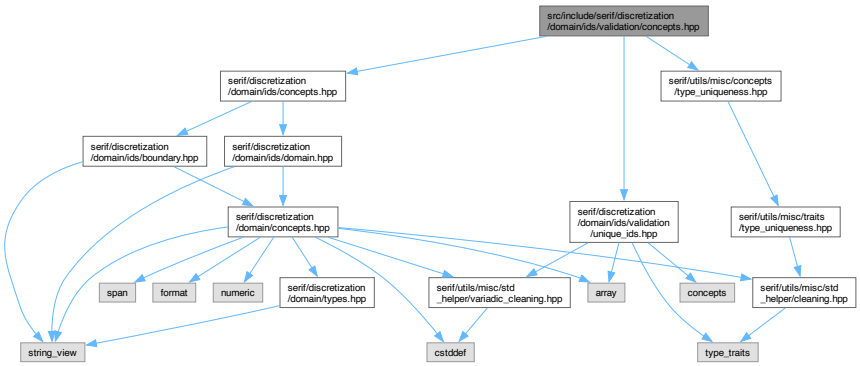
Author

Emily M. Boudreaux

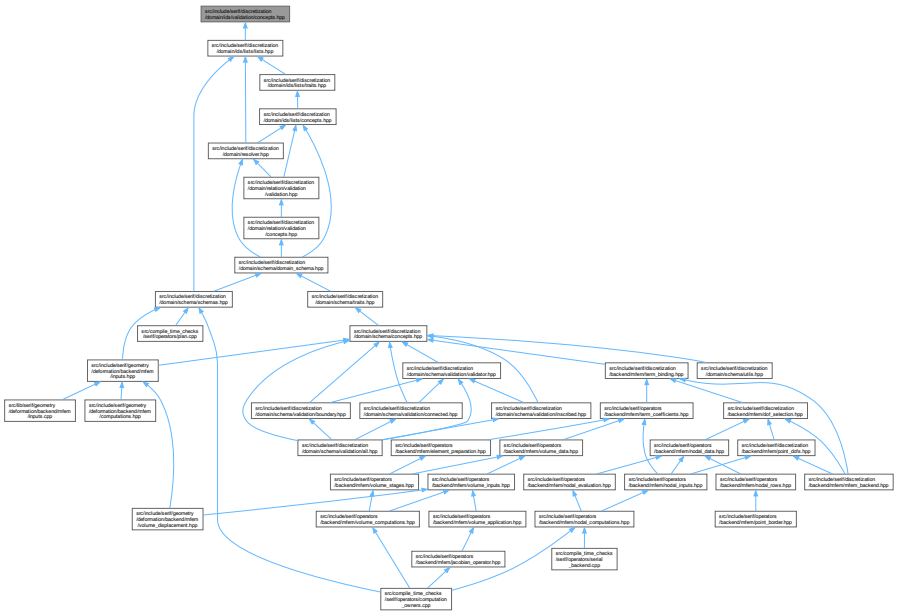
11.183 src/include/serif/discretization/domain/ids/validation/concepts.hpp File Reference

This file defines concepts for domain ID lists in the discretization framework. Specifically we define here concepts used to validate traits of a domain.

#include "serif/discretization/domain/ids/concepts.hpp"
#include "serif/discretization/domain/ids/validation/unique_ids.hpp"
#include "serif/utils/misc/concepts/type_uniqueness.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::ids
- namespace serif::discretization::domain::ids::validation

Concepts

- concept serif::discretization::domain::ids::validation::HaveUniqueDomainIDs

  Concept to check if a pack of domain IDs have unique ID values. This is used to ensure that no two domain IDs in a list have the same ID.
- concept serif::discretization::domain::ids::validation::HaveUniqueBoundaryIDs

  Concept to check if a pack of boundary IDs have unique ID values. This is used to ensure that no two boundary IDs in a list have the same ID.
- concept serif::discretization::domain::ids::validation::HaveUniqueDomains

  Concept to check if a pack of domain IDs have unique domain types. This is used to ensure that no two domain IDs in a list refer to the same domain type.
- concept serif::discretization::domain::ids::validation::HaveUniqueBoundaries

  Concept to check if a pack of boundary IDs have unique boundary types. This is used to ensure that no two boundary IDs in a list refer to the same boundary type.
- concept serif::discretization::domain::ids::validation::AreDistinctDomainIDs

  Concept to check if a pack of domain IDs are distinct. This means that each ID is a valid domain ID, no two IDs have the same integer value, and no two IDs refer to the same domain type.
- concept serif::discretization::domain::ids::validation::AreDistinctBoundaryIDs

  Concept to check if a pack of boundary IDs are distinct. This means that each ID is a valid boundary ID, no two IDs have the same integer value, and no two IDs refer to the same boundary type.

11.183.1 Detailed Description

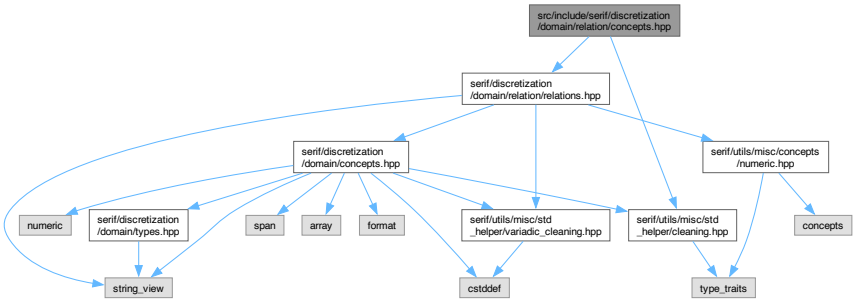
This file defines concepts for domain ID lists in the discretization framework. Specifically we define here concepts used to validate traits of a domain.

Author

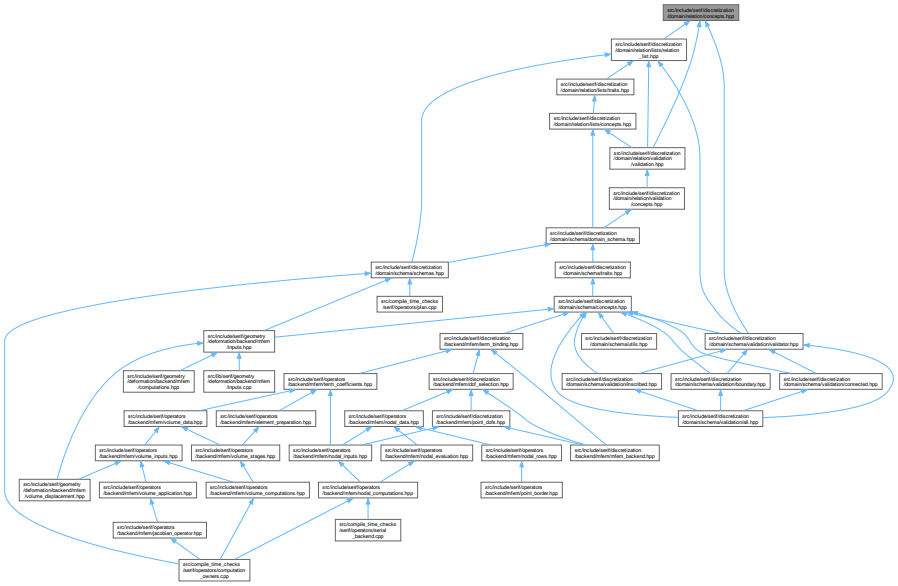
Emily M. Boudreaux

11.184 src/include/serif/discretization/domain/relation/concepts.hpp File Reference

#include "serif/discretization/domain/relation/relations.hpp"
#include "serif/utils/misc/std_helper/cleaning.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::relation

Concepts

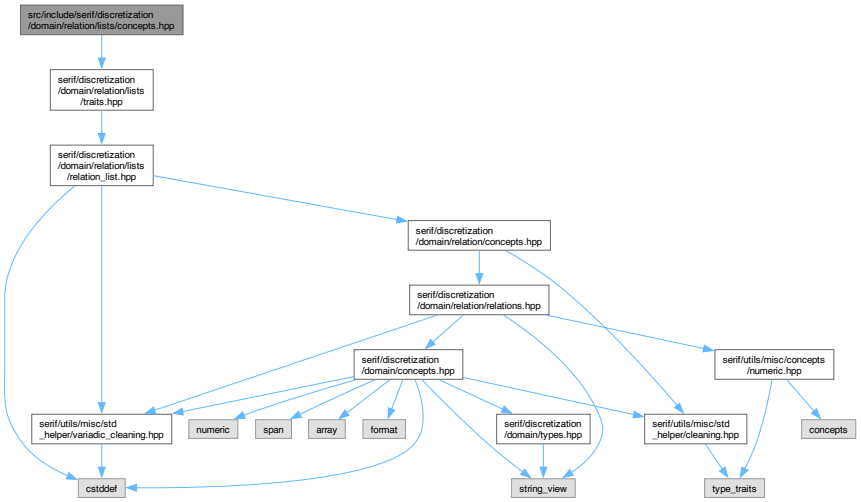
- concept serif::discretization::domain::relation::IsRelation

Variables

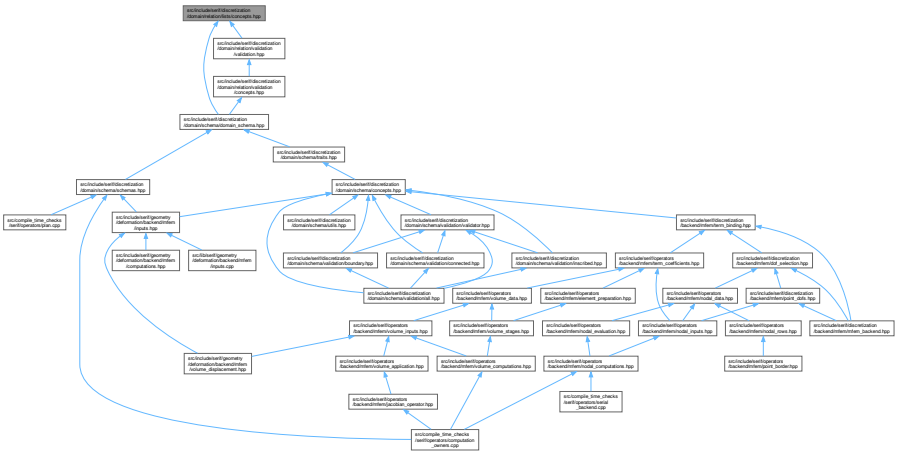
- template<class Child, class Parent>
  constexpr bool serif::discretization::domain::relation::AIsBaseClassOfB

11.185 src/include/serif/discretization/domain/relation/lists/concepts.hpp File Reference

#include "serif/discretization/domain/relation/lists/traits.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

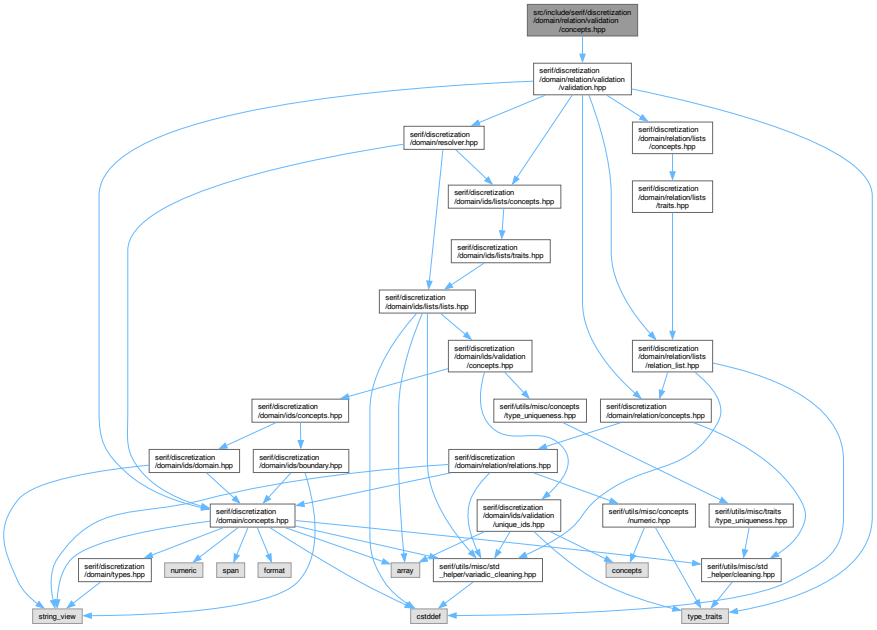
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::relation
- namespace serif::discretization::domain::relation::lists

Concepts

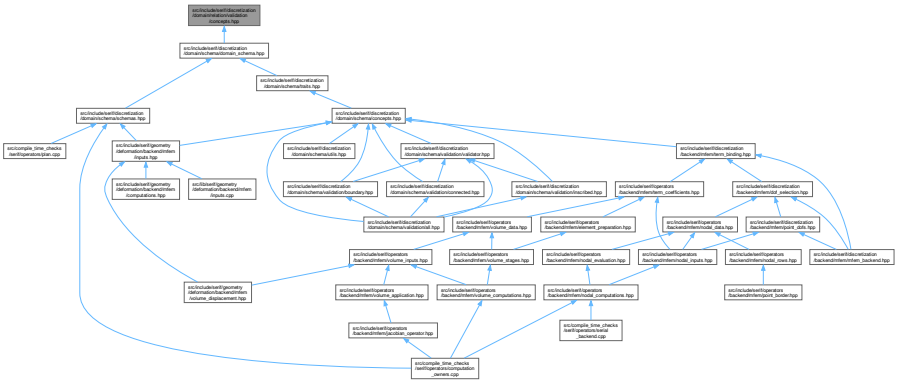
- concept serif::discretization::domain::relation::lists::IsRelationList

11.186 src/include/serif/discretization/domain/relation/validation/concepts.hpp File Reference

#include "serif/discretization/domain/relation/validation/validation.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::relation
- namespace serif::discretization::domain::relation::validation

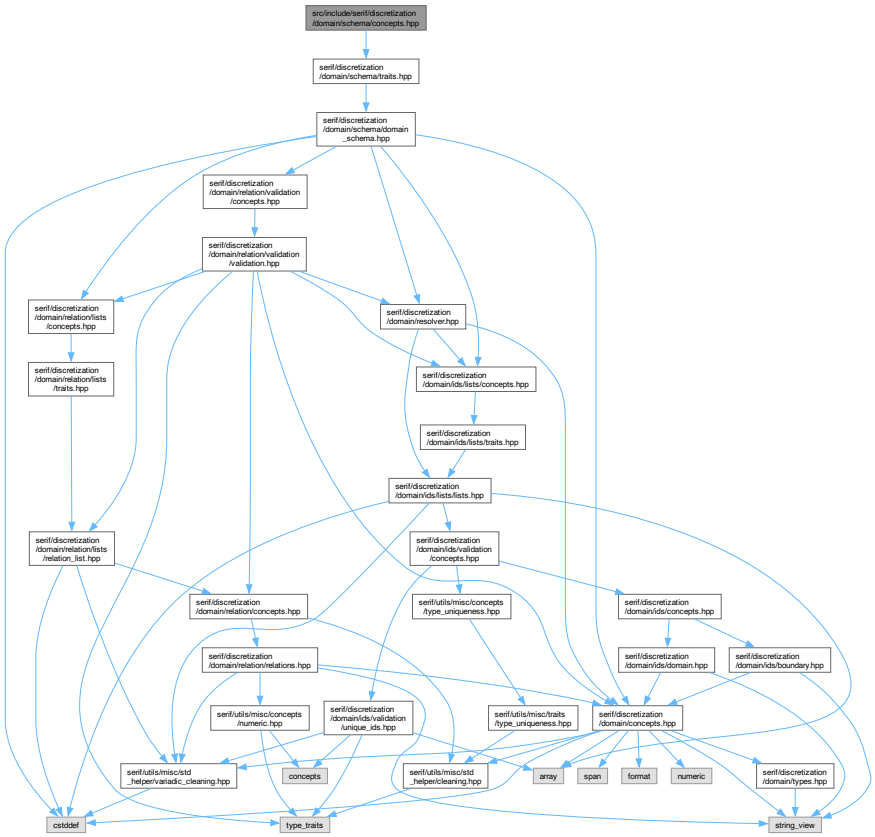
Concepts

- concept serif::discretization::domain::relation::validation::HaveValidRelationEntities

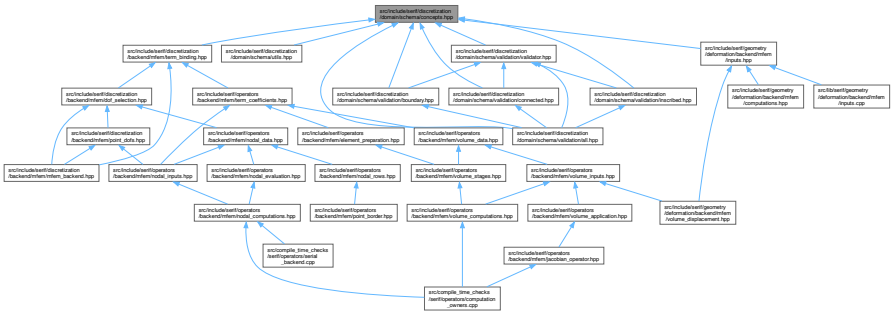
11.187 src/include/serif/discretization/domain/schema/concepts.hpp File Reference

#include "serif/discretization/domain/schema/traits.hpp"

Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

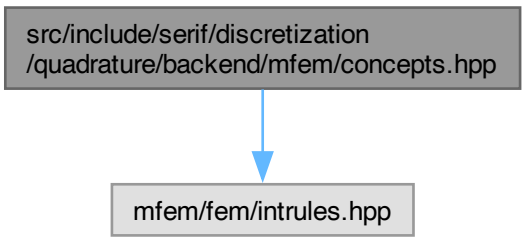
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema

Concepts

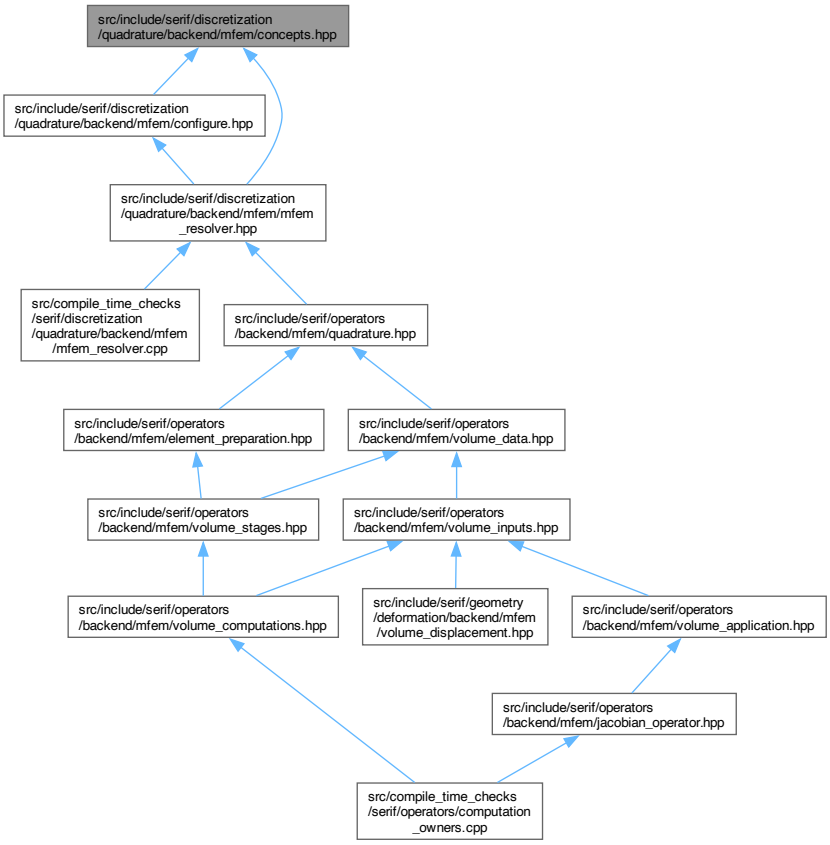
- concept serif::discretization::domain::schema::IsSchema

11.188 src/include/serif/discretization/quadrature/backend/mfem/concepts.hpp File Reference

#include "mfem/fem/intrules.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

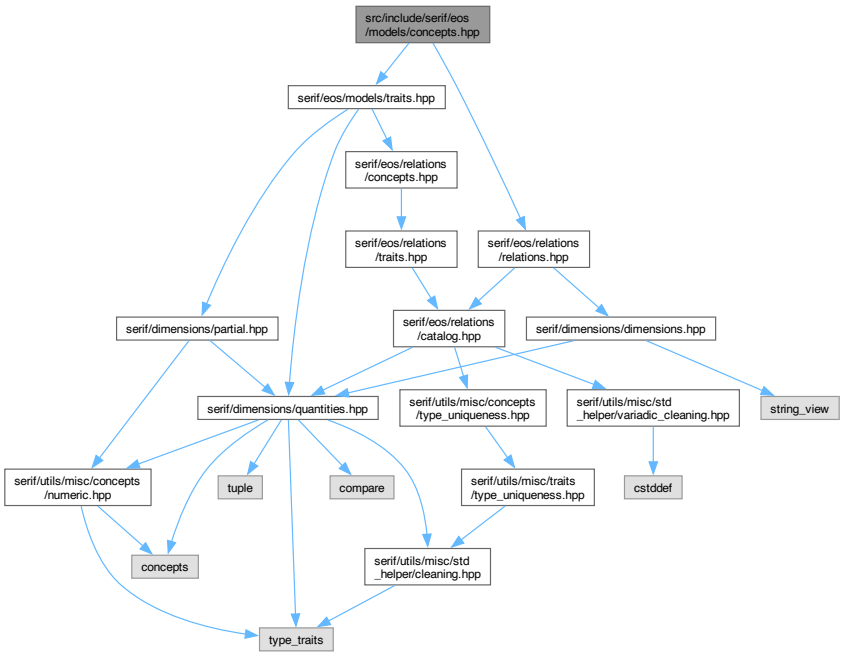
- namespace serif
- namespace serif::discretization
- namespace serif::discretization::quadrature
- namespace serif::discretization::quadrature::backend
- namespace serif::discretization::quadrature::backend::mfem

Concepts

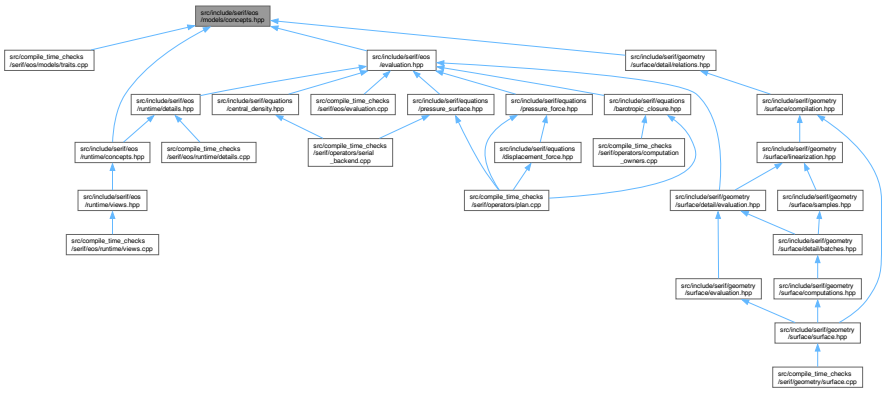
- concept serif::discretization::quadrature::backend::mfem::HasReferenceSetIntegrationRule
- concept serif::discretization::quadrature::backend::mfem::HasPointerSetIntegrationRule
- concept serif::discretization::quadrature::backend::mfem::HasSetIntRule
- concept serif::discretization::quadrature::backend::mfem::ConfigurableIntegrator

11.189 src/include/serif/eos/models/concepts.hpp File Reference

#include "serif/eos/models/traits.hpp"
#include "serif/eos/relations/relations.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

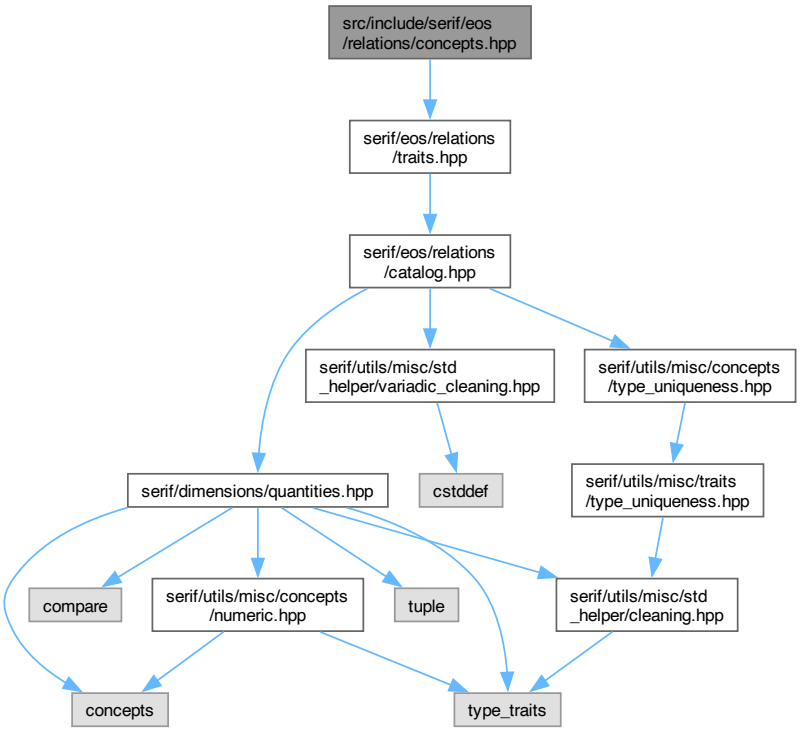
- namespace serif
- namespace serif::eos
- namespace serif::eos::models

Concepts

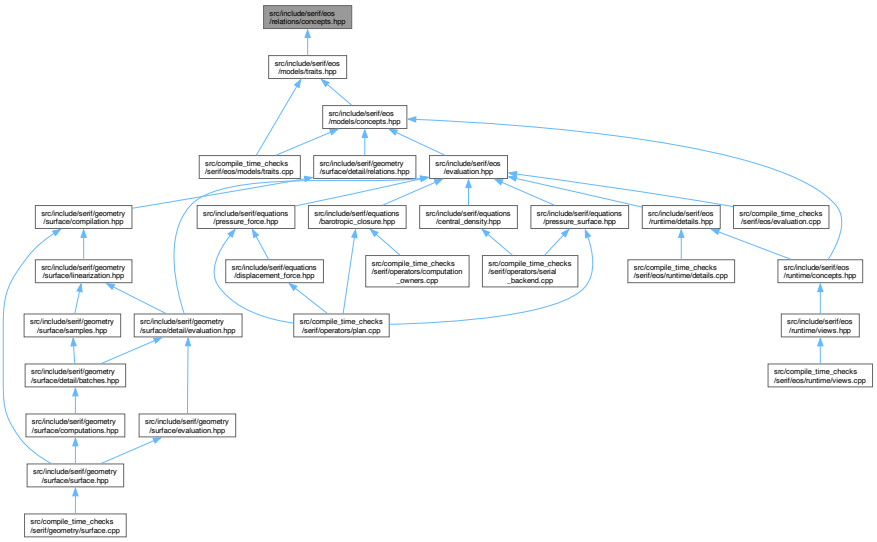
- concept serif::eos::models::EOSModel
- concept serif::eos::models::EOSSupportingRelation
- concept serif::eos::models::EOSSupportingPartialDerivative
- concept serif::eos::models::StructureSeedEOS
- concept serif::eos::models::BarotropicClosureEOS
- concept serif::eos::models::PressureForceEOS

11.190 src/include/serif/eos/relations/concepts.hpp File Reference

#include "serif/eos/relations/traits.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::eos
- namespace serif::eos::relations

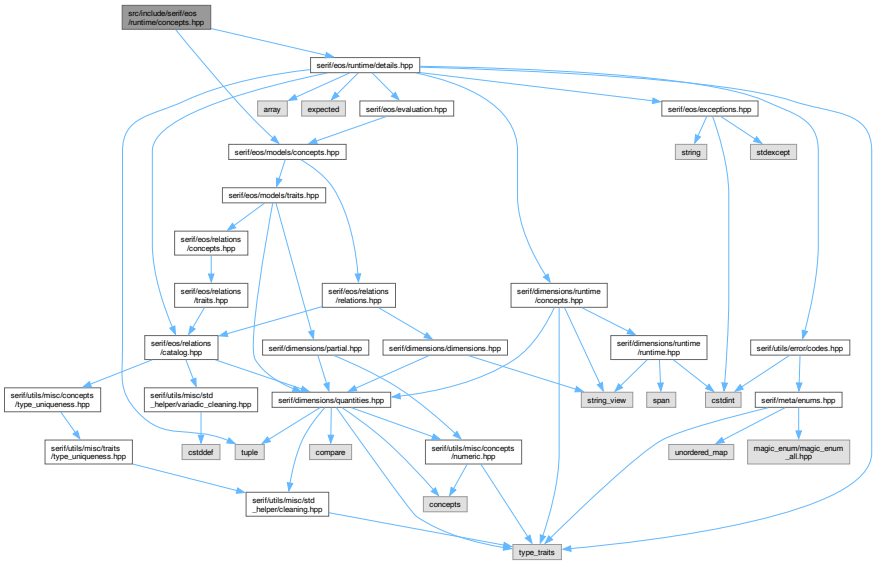
Concepts

- concept serif::eos::relations::CatalogContainingRelation
- concept serif::eos::relations::RelationContainingInput
- concept serif::eos::relations::ValidThermodynamicRelation
- concept serif::eos::relations::ValidRelationCatalog

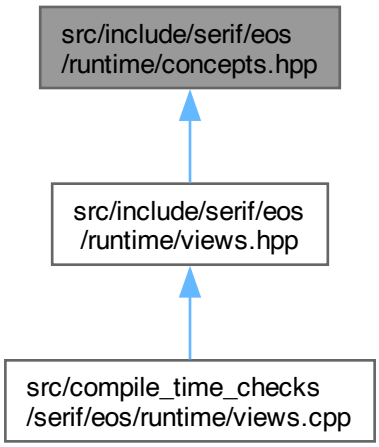
11.191 src/include/serif/eos/runtime/concepts.hpp File Reference

#include ”serif/eos/models/concepts.hpp”
#include ”serif/eos/runtime/details.hpp”

Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::eos
- namespace serif::eos::runtime

Concepts

- concept serif::eos::runtime::RuntimeEOSModel

Typedefs

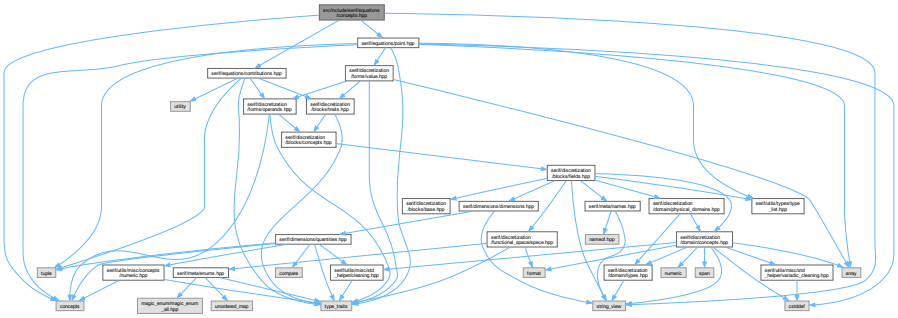
- template<RuntimeEOSModel EOSCandidate>
  using serif::eos::runtime::RuntimeEOSAdapter = RuntimeCatalogDispatch<EOSCandidate, typename EOSCandidate::Relations>

Functions

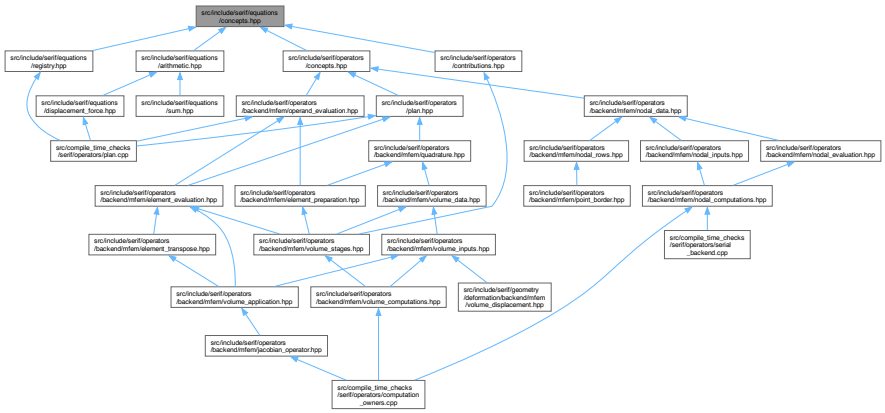
- template<RuntimeEOSModel EOSCandidate>
  constexpr std::span< const dimensions::runtime::RuntimeRelationDescriptor > serif::eos::runtime::runtimeRelationDesc
  () noexcept

11.192 src/include/serif/equations/concepts.hpp File Reference

#include <concepts>
#include <string_view>
#include "serif/equations/contributions.hpp"
#include "serif/equations/point.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::equations::ValidContributions< RowT, TupleT >
- struct serif::equations::ValidContributions< RowT, std::tuple< C... > >

Namespaces

- namespace serif
- namespace serif::equations

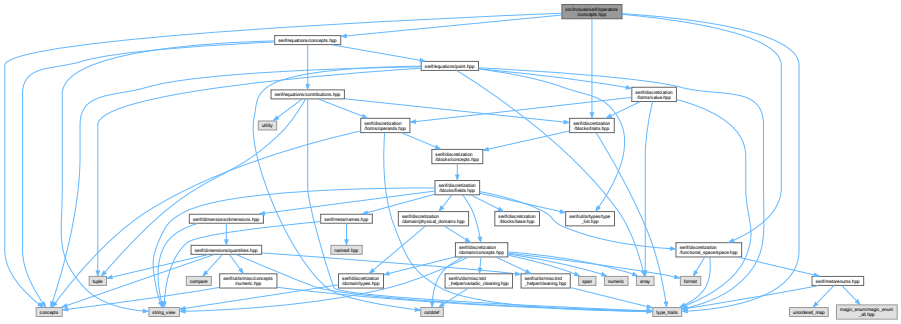
Concepts

- concept serif::equations::PointEquation
- concept serif::equations::Equation

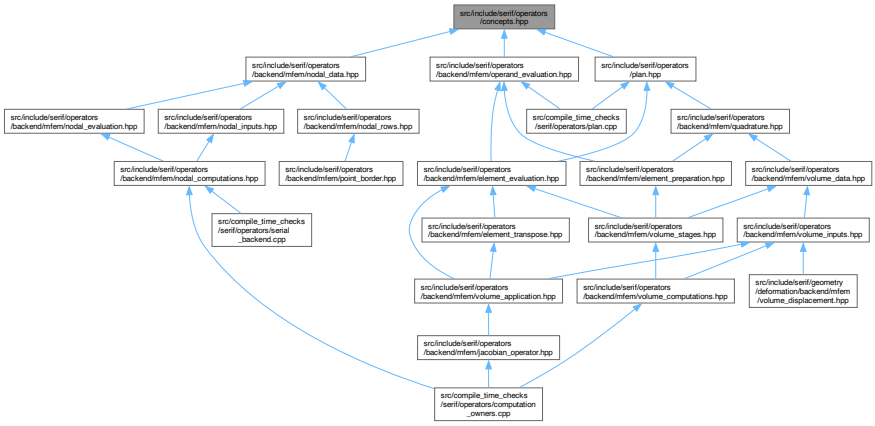
  Physics authoring contract; contributions share one residual row but may use different tests/supports.

11.193 src/include/serif/operators/concepts.hpp File Reference

#include <concepts>
#include <type_traits>
#include "serif/discretization/blocks/traits.hpp"
#include "serif/discretization/functional_space/space.hpp"
#include "serif/equations/concepts.hpp"
Include dependency graph for concepts.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::HasScalarVolumeOperands< EquationT, OperandsT >
- struct serif::operators::HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > >
- struct serif::operators::VolumeOperands< C, ListT >
- struct serif::operators::VolumeOperands< C, utils::types::TypeList< O... > >
- struct serif::operators::VolumeContributions< TupleT >
- struct serif::operators::VolumeContributions< std::tuple< C... > >

Namespaces

- namespace serif
- namespace serif::operators

Concepts

- concept serif::operators::NodalValueEquation

  Nodal execution: an H1 value depending only on itself, with no explicit position dependence.
- concept serif::operators::ScalarValueTerm
- concept serif::operators::ScalarValueOperand
- concept serif::operators::ScalarVolumeEquation

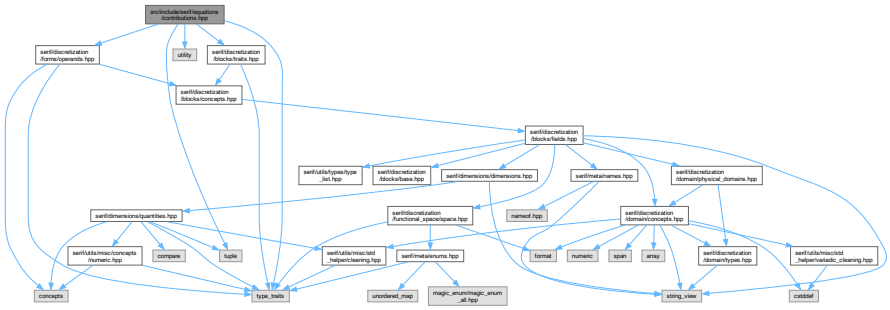
  Equations executable by the current scalar-value volume operator.
- concept serif::operators::VolumeOperand

  Mathematical operand combinations implemented by the current volume backend.
- concept serif::operators::VolumeEquation

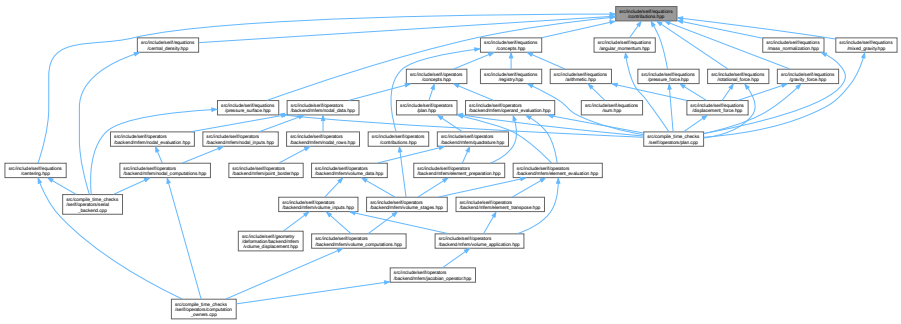
  One or more supported weak volume contributions assembled into a single residual row.

11.194 src/include/serif/equations/contributions.hpp File Reference

#include <tuple>
#include <type_traits>
#include <utility>
#include "serif/discretization/blocks/traits.hpp"
#include "serif/discretization/forms/operands.hpp"
Include dependency graph for contributions.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::equations::Contribution< TestOperandT, SupportT >

  Optional authoring base: a test operation and the domains of one weak contribution.
- struct serif::equations::TestOperandTraits< T, typename >
- struct serif::equations::TestOperandTraits< T, std::void_t< typename T::TestOperand > >
- struct serif::equations::ContributionSupport< T, typename >
- struct serif::equations::ContributionSupport< T, std::void_t< typename T::Support > >
- struct serif::equations::Contributions< T, typename >

  Collect leaf equation types without copying the objects in a composition.
- struct serif::equations::ContributionTuple< TupleT, typename >
- struct serif::equations::ContributionTuple< std::tuple< Ts... >, std::void_t< decltype(std::tuple_cat(std::declval< typ
- struct serif::equations::Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions())> >

Namespaces

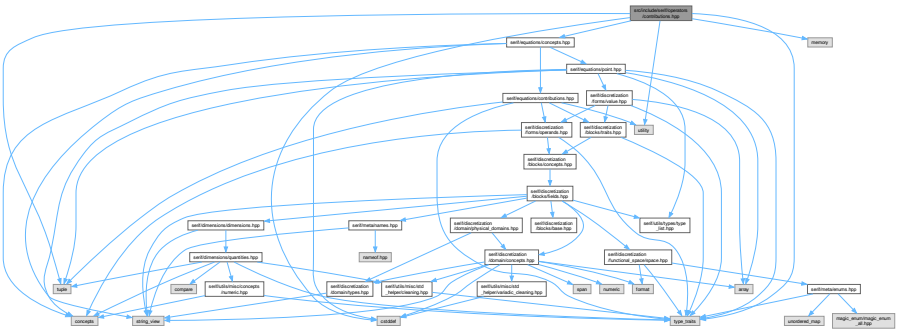
- namespace serif
- namespace serif::equations

Typedefs

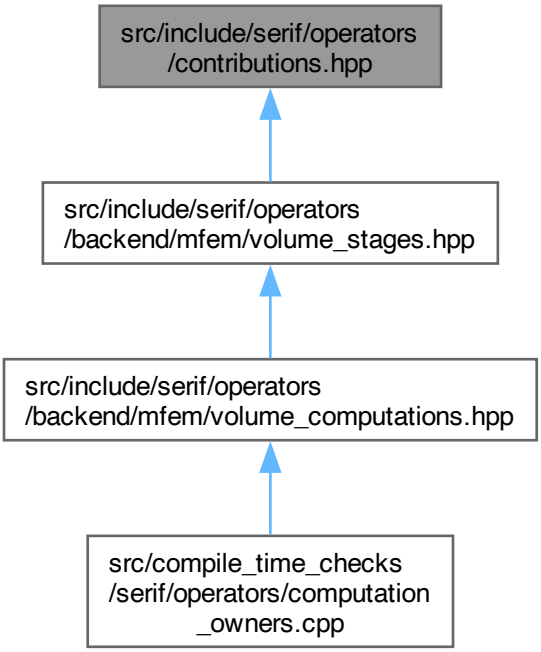
- template<typename T>
  using serif::equations::TestOperandOf = typename TestOperandTraits<T>::type
- template<typename T>
  using serif::equations::SupportOf = typename ContributionSupport<T>::type
- template<typename T>
  using serif::equations::ContributionsOf = typename Contributions<std::remove_cvref_t<T>>::type

11.195 src/include/serif/operators/contributions.hpp File Reference

#include <cstddef>
#include <memory>
#include <type_traits>
#include <utility>
#include <tuple>
#include "serif/equations/concepts.hpp"
Include dependency graph for contributions.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::ContributionBinding< EquationT >

  Retain one leaf equation and the constant coefficient of its arithmetic expression.
- struct serif::operators::detail::ContributionChildren< EquationT >

Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::detail

Functions

- template<typename EquationT>
  auto serif::operators::detail::bind_contributions (std::shared_ptr< const EquationT > equation, double coefficient)

  Flatten equation arithmetic while preserving every generated tuple and borrowed parent.

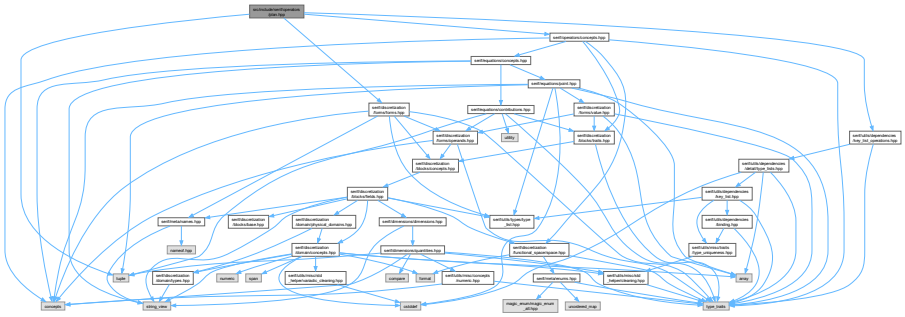
- template<typename EquationT, typename FunctionT>
  void serif::operators::detail::visit_contributions (const EquationT &equation, FunctionT &function, std::size_t &index, double coefficient)

  Visit arithmetic leaves without retaining them beyond the supplied callback.
- template<equations::Equation EquationT>
  auto serif::operators::bind_contributions (std::shared_ptr< const EquationT > equation)

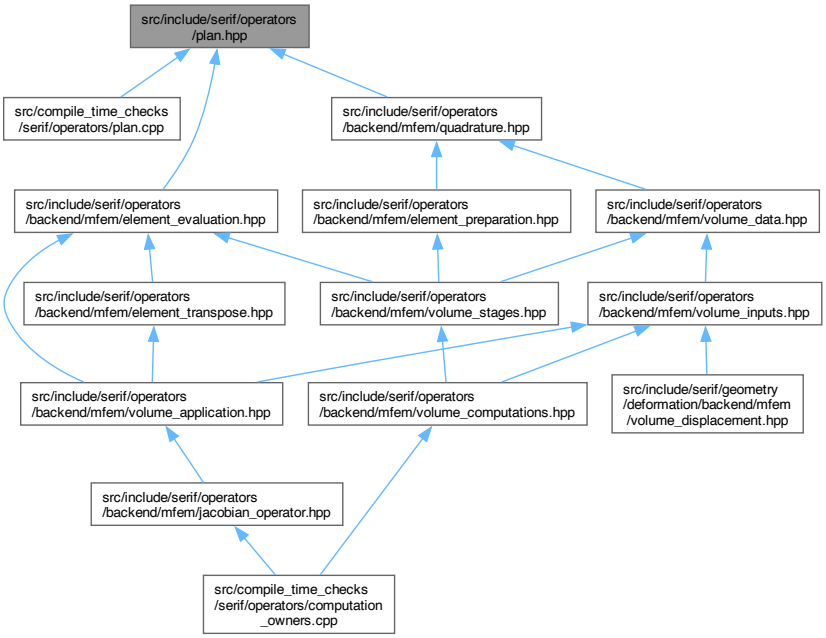
  Give each retained leaf a compile-time tuple position for independently cached contribution results.
- template<equations::Equation EquationT, typename FunctionT>
  void serif::operators::for_each_contribution (const EquationT &equation, FunctionT &&function)

11.196 src/include/serif/operators/plan.hpp File Reference

#include <tuple>
#include "serif/utils/dependencies/key_list_operations.hpp"
#include "serif/discretization/forms/forms.hpp"
#include "serif/operators/concepts.hpp"
Include dependency graph for plan.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::operators::detail::AsKeyList< utils::types::TypeList< T... > >
- struct serif::operators::detail::Join< Lists >
- struct serif::operators::detail::ContributionOperands< std::tuple< C... > >
- struct serif::operators::detail::Columns< utils::types::TypeList< O... > >
- struct serif::operators::detail::CouplingForm< Input, TestOperand >
- struct serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >

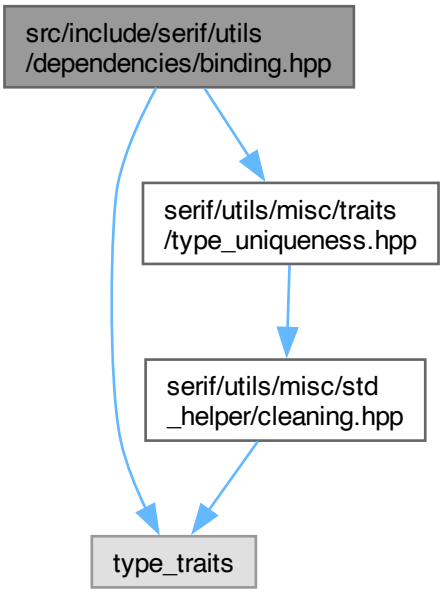
Namespaces

- namespace serif
- namespace serif::operators
- namespace serif::operators::detail

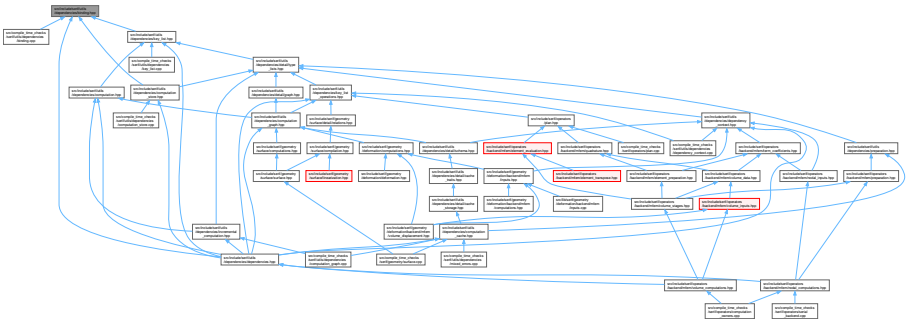
11.197 src/include/serif/utils/dependencies/binding.hpp File Reference

#include <type_traits>
#include "serif/utils/misc/traits/type_uniqueness.hpp"

Include dependency graph for binding.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::Binding< KeyT, TargetT >
- struct serif::utils::dependencies::Bindings< BindingTs >
- struct serif::utils::dependencies::detail::HasBinding< Bindings< BindingTs... >, KeyT >
- struct serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >
- struct serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< K, T >
- struct serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::collector
- struct serif::utils::dependencies::HasBinding< BindingsT, KeyT >
- struct serif::utils::dependencies::BoundType< BindingsT, KeyT >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies
- namespace serif::utils::dependencies::detail

Concepts

- concept serif::utils::dependencies::IsBinding
- concept serif::utils::dependencies::IsBindings
- concept serif::utils::dependencies::BindingsWithBinding

Typedefs

- template<typename BindingsT, typename KeyT>
  using serif::utils::dependencies::bound_type_t = BoundType<BindingsT, KeyT>::target_type

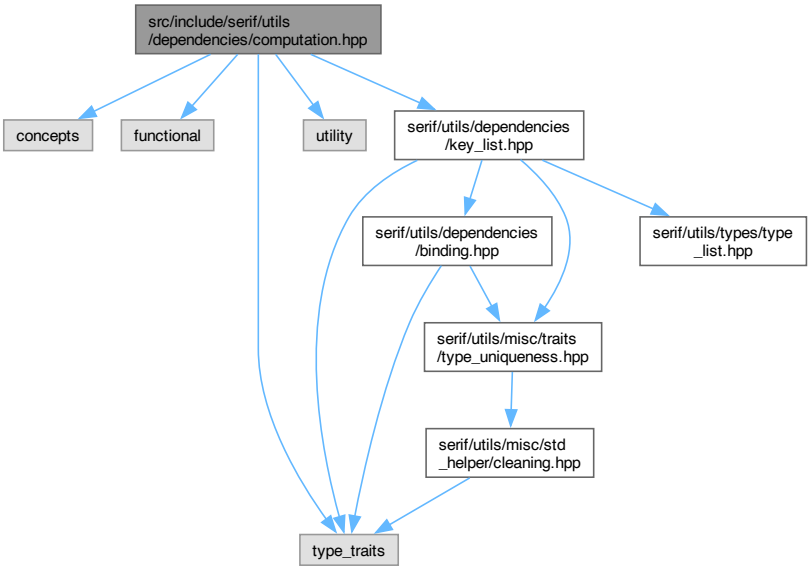
Variables

- template<typename T>
  constexpr bool serif::utils::dependencies::is_bindings_v = false
- template<IsBinding... BindingTs>
  constexpr bool serif::utils::dependencies::is_bindings_v< Bindings< BindingTs... > > = misc::traits::all_unique_types
  BindingTs::Key...>
- template<IsBindings BindingsT, typename KeyT>
  constexpr bool serif::utils::dependencies::has_binding_v = HasBinding<BindingsT, KeyT>::value

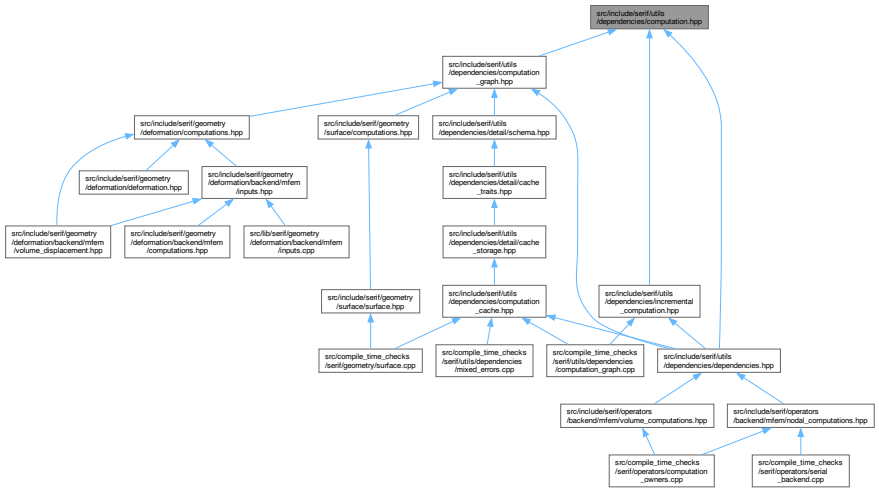
11.198 src/include/serif/utils/dependencies/computation.hpp File Reference

#include <concepts>
#include <functional>
#include <type_traits>
#include <utility>

#include "serif/utils/dependencies/key_list.hpp"
Include dependency graph for computation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

Concepts

- concept serif::utils::dependencies::IsComputation

Functions

- template<typename KeyT, IsKeyList InputsT, typename FunctionT>
  requires std::constructible_from<std::decay_t<FunctionT>, FunctionT>
  auto serif::utils::dependencies::make_computation (FunctionT &&function)

  Attach an output key and ordered input keys to a lambda, function, or other callable.

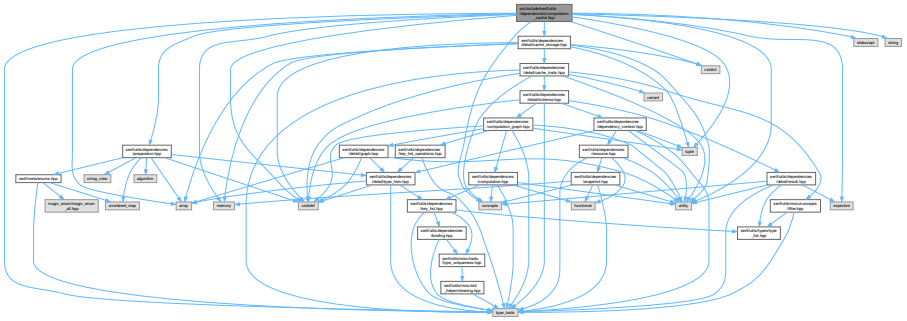
Variables

- template<typename T>
  constexpr bool serif::utils::dependencies::is_computation_v = false
- template<typename KeyT, typename InputsT, typename FunctionT>
  constexpr bool serif::utils::dependencies::is_computation_v< Computation< KeyT, InputsT, FunctionT > >
  = true

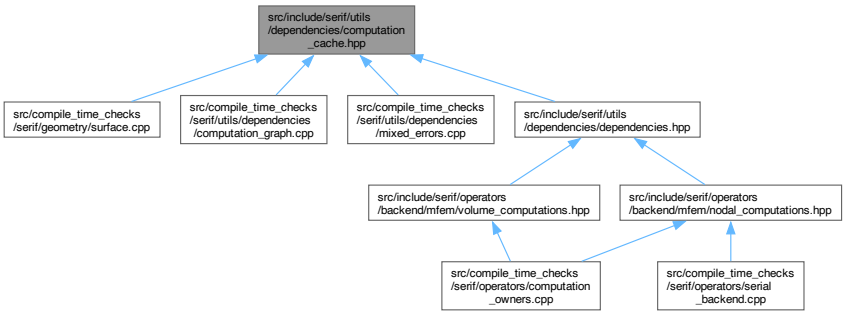
11.199 src/include/serif/utils/dependencies/computation_cache.hpp File Reference

#include <array>
#include <concepts>
#include <cstddef>
#include <cstdint>
#include <expected>
#include <memory>
#include <stdexcept>
#include <string>
#include <tuple>
#include <type_traits>
#include <utility>
#include "serif/utils/dependencies/preparation.hpp"

#include "serif/utils/dependencies/detail/cache_storage.hpp"
Include dependency graph for computation_cache.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

Functions

- template<typename ErrorT = std::string, IsComputationGraph GraphT, IsDependencyContext ContextT>
  requires CanPrepare<GraphT, ContextT, ErrorT> && std::constructible_from<std::remove_cvref_t<GraphT>, GraphT>
  auto serif::utils::dependencies::make_computation_cache (GraphT &&graph, const ContextT &)

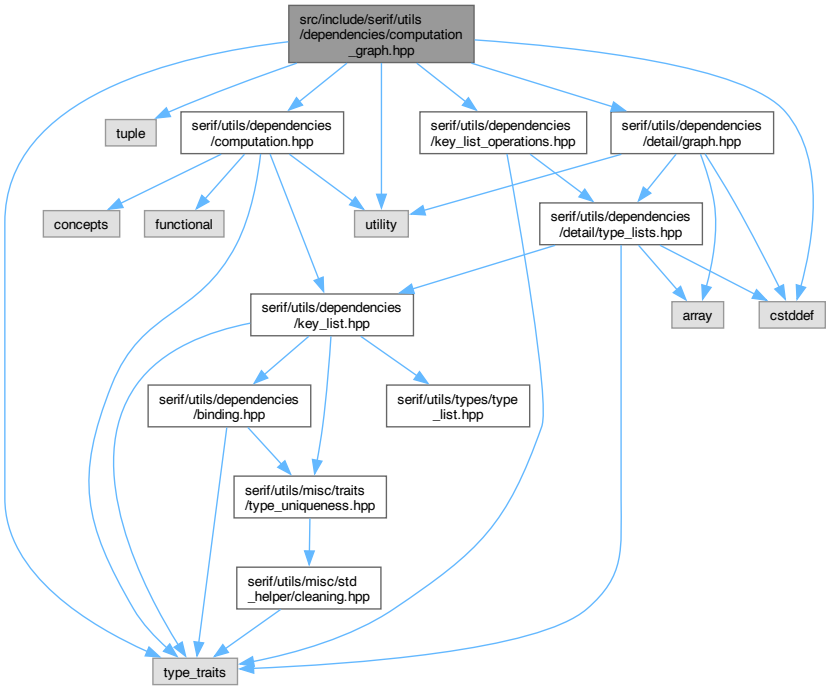
  Fix the graph's source types without retaining the supplied context or evaluating any computations.

- template<typename ErrorT = std::string, IsDependencyContext ContextT, typename... Computations> requires CompatibleComputations<Computations...> && (std::constructible_from<std::remove_cvref_t<Computations>, Computations> && ...) && CanPrepare<ComputationGraph<std::remove_cvref_t<Computations>...>, ContextT, ErrorT>
  auto serif::utils::dependencies::make_computation_cache (const ContextT &context, Computations &&... computations)

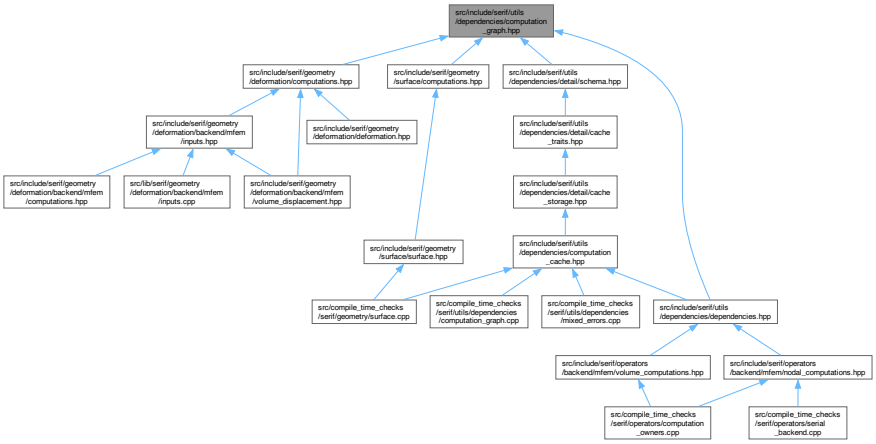
  Build a checked graph and an empty cache directly from named computations.

11.200 src/include/serif/utils/dependencies/computation_graph.hpp File Reference

#include <cstddef>
#include <tuple>
#include <type_traits>
#include <utility>
#include "serif/utils/dependencies/computation.hpp"
#include "serif/utils/dependencies/key_list_operations.hpp"
#include "serif/utils/dependencies/detail/graph.hpp"
Include dependency graph for computation_graph.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::utils::dependencies::ComputationGraph< Computations >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

Concepts

- concept serif::utils::dependencies::CompatibleComputations
- concept serif::utils::dependencies::IsComputationGraph

Functions

- template<typename... Computations>
  requires CompatibleComputations<Computations...> && (std::constructible_from<std::remove_cvref_t<Computations>, Computations> && ...)
  auto serif::utils::dependencies::make_computation_graph (Computations &&... computations)

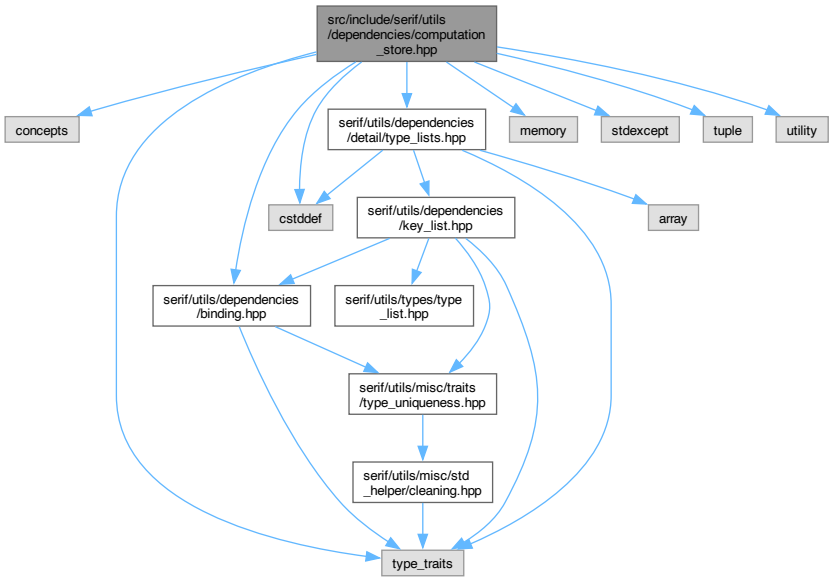
  Reject duplicate producers and cyclic preparation dependencies at the declaration site.

Variables

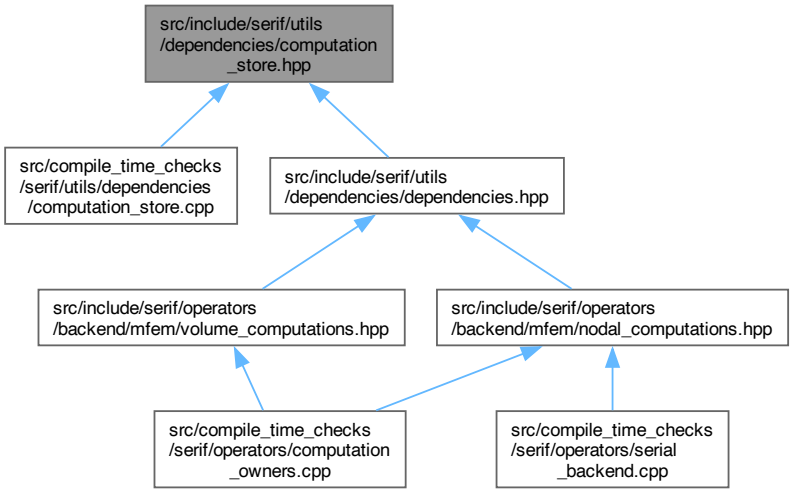
- template<typename T>
  constexpr bool serif::utils::dependencies::is_computation_graph_v = false
- template<typename... Computations>
  constexpr bool serif::utils::dependencies::is_computation_graph_v< ComputationGraph< Computations... > >
  = true

11.201 src/include/serif/utils/dependencies/computation_store.hpp File Reference

#include <concepts>
#include <cstddef>
#include <memory>
#include <stdexcept>
#include <tuple>
#include <type_traits>
#include <utility>
#include "serif/utils/dependencies/binding.hpp"
#include "serif/utils/dependencies/detail/type_lists.hpp"
Include dependency graph for computation_store.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >
- class serif::utils::dependencies::ComputationStore< EntryTs >

  Own independent computation caches under compile-time role keys; this store does not connect their graphs.

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

Concepts

- concept serif::utils::dependencies::IsOwnedComputation

Functions

- template<typename KeyT, typename ComputationT>
  requires std::constructible_from<std::remove_cvref_t<ComputationT>, ComputationT>
  auto serif::utils::dependencies::own_computation (ComputationT &&computation)

  Name an owned computation; rvalues transfer ownership and copyable lvalues are copied.
- template<typename... EntryTs>
  requires (IsOwnedComputation<EntryTs> && ...) && misc::traits::all_unique_types_v<typename std::remove_cvref_t<EntryTs>::Key...> && (std::constructible_from<std::remove_cvref_t<EntryTs>, EntryTs> && ...)
  auto serif::utils::dependencies::make_computation_store (EntryTs &&... entries)

  Construct the model's computation owner, rejecting duplicate role keys before execution.

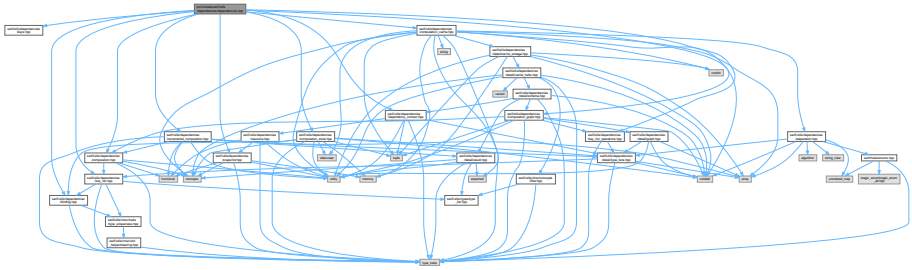
Variables

- template<typename T>
  constexpr bool serif::utils::dependencies::is_owned_computation_v = false
- template<typename KeyT, typename ComputationT>
  constexpr bool serif::utils::dependencies::is_owned_computation_v< OwnedComputation< KeyT, ComputationT > >
  = true

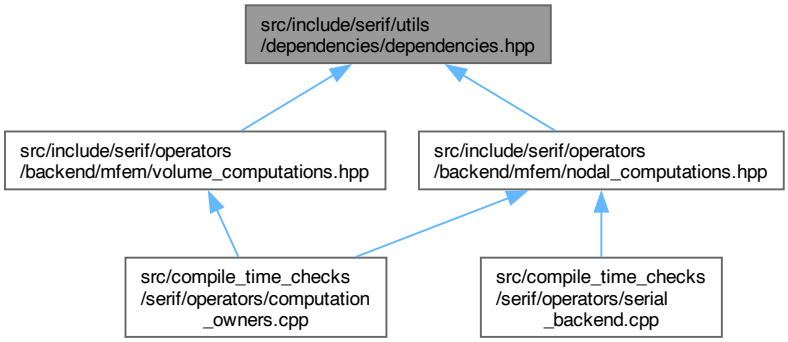
11.202 src/include/serif/utils/dependencies/dependencies.hpp File Reference

This file is a convenience header which includes all of the headers in the dependencies module. Further, we include the detailed documentation for the dependencies module here.

#include "serif/utils/dependencies/keys.hpp"
#include "serif/utils/dependencies/binding.hpp"
#include "serif/utils/dependencies/key_list.hpp"
#include "serif/utils/dependencies/key_list_operations.hpp"
#include "serif/utils/dependencies/dependency_context.hpp"
#include "serif/utils/dependencies/computation.hpp"
#include "serif/utils/dependencies/incremental_computation.hpp"
#include "serif/utils/dependencies/computation_graph.hpp"
#include "serif/utils/dependencies/computation_cache.hpp"
#include "serif/utils/dependencies/computation_store.hpp"
#include "serif/utils/dependencies/snapshot.hpp"
Include dependency graph for dependencies.hpp:



This graph shows which files directly or indirectly include this file:



11.202.1 Detailed Description

This file is a convenience header which includes all of the headers in the dependencies module. Further, we include the detailed documentation for the dependencies module here.

Author

Emily M. Boudreaux

Dependencies Module

Demonstration code for how to use the SERiF computational graph system The broad idea here is that there are many calculations which depend on each other but which may not necessarily always need to be recomputed just because another part of the model has changed. We model this as a combination of a acyclic computational graph, defined at compile time, and a runtime cache of values. The general workflow is that a developer defines the ”context” of the computational graph, think of this as all of the inputs to the graph, or maybe more specifically, anything that could change due to some external event (e.g. a time step, a newton step, etc...) This context is probably the most CS intensive part of this code as the developer needs to think about lifetimes and ownership.

Context Contexts are made with the make_dependency_context function, this accepts a variadic pack of bindings between a type and a runtime value. You can generate these yourself using the binding functions and types; however, we provide two convince wrappers which generally developers should consider using.

- borrow: Used to bind a type to a reference to a value. This is useful for values which are owned by the caller and which the caller will manage the lifetime of (takes an lvalue).
- own: Used to bind a type to a value which is owned by the context (takes an rvalue) Each of these functions accepts a type as a template parameter and a runtime value as an argument. The type is used as an ID, this is key to understand, if we write

```
using namespace serif::utils::dependencies;

double x{1.0};
auto context = make_dependency_context(borrow<keys::Input>(x));
```

we are not saying that x is of type keys::Input, this is clearly not the case. Rather, we are registering in the computational graph a binding between the type keys::Input and the runtime value x. Think of this as naming the various inputs to the graph.

Computations After defining a context we can define a computation, computations define both nodes and edges in the computational graph (nodes by the body, edges by the template parameters). Consider the following example (assuming the context from above):

```
auto func = make_computation<keys::Output, InputList<keys::Input>>(
 [](const double _x) -> double {
    return _x * _x;
 }
)
```

what we have defined here is a node which takes a single input tagged keys::Input and produces an output tagged as keys::Output. Note how we have implicitly registerd a new edge on the graph (even though its not in the context). This is an internal edge which cannot be accessed by the user. Rather, these internal edges allow us to connect computations together to form a larger graph. Consider now the following code

```
using namespace serif::utils::dependencies;

struct tags {
   struct X{};
   struct Y{};

   struct Output1{};
   struct Output2{};
};

double x{1.0};
double y{2.0};
auto context = make_dependency_context(borrow<tags::X>(x), borrow<tags::Y>(y));

auto func1 = make_computation<tags::Output1, InputList<tags::X>>(
   [](const double _x) -> double {
      return _x * _x;
   }
)

auto func2 = make_computation<tags::Output2, InputList<tags::Output1, tags::Y>>(
  [](const double _x, const double _y) -> double {
      return _x + _y;
   }
);
```

we have now defined two computations, the first takes a single input, whichever input is tagged as tags::X and produces an output tagged as tags::Output1. The second computation takes two inputs, the first is the output of the first computation (lookup by the tag) and the second is whichever input is tagged as tags::Y, and produces an output tagged as tags::Output2. Note that the order of arguments to the lambda must be the same as the order of the tags in the InputList, the dependencies module will automatically order arguments based on the InputList. There is type checking to ensure that the types match, but if you have two inputs of the same type that are flipped there is no way for the system to inspect that and warn you. Further, depending on your compiler and flags there may be implicit casting between convertable types (e.g. int <-> double).

Todo We may want to remove the ability to have implicit casting between types, this is a bit of a footgun and may lead to subtle bugs.

Note

This is primary goal of this system, split computations into small pieces with the ability to cache intermediate results and reuse as many of those intermediate results as we can can prove is safe to do. The whole reason this system is so complex is that very purpose. There is a fundamental problem with high dimensional stellar models, they are always going to be slow, therefore we need to accept some degree of complexity in order to make them tractable.

Computation Cache Once we have the computations we need to connect them together, generally use the helper function make_computational_cache which takes a context and a variadic pack of computations. There are a lot of compile time checks which run to ensure that the graph is valid, the context describes what is needed to compute the graph, and that each computation is valid. Generally if a static analysis error or compile error pops up here it probably means there is a mistake in the definition of the context or the computations. I've tried to make these errors as clear as possible; however, there is only so much that you can do with the C++ compiler. I recommend first checking, probably with a drawn-out diagram, that all the tags are correct and what you expect.

A developer can also choose to use the make_computational_graph function which provides a bit more explicit control; however, there is generally very little reason to do this and that function is primarily intended for internal use (for those interested make_computational_cache just calls make_computational_graph and then binds the particular context to the graph). Now consider the following code assuming the context and computations from above:

```
auto graph_cache = make_computation_cache(graph_context, quadratic_function, linear_function);
```

At this point we have a cache which we can access various parts of; however, no computations have been run yet. To run a computation we must call try_prepare.

Todo We really should remove the try_prepare'dness of it all, that function uses exceptions for runtime logic which would make Herb Sutter very mad (and on a more serious note can lead to significant performance issues as the stack is unwound). This is a larger refactor as we need to propegate value based error handeling all through the code but should be done at some point.

try_prepare takes the context and attempts to evaluate the graph using the minimum number of nodal evaluations necessary. This means the first time, since nothing has been cached yet, the entire graph will be evaluated.

```
auto report = graph_cache.try_prepare(graph_context);

double func1_result = graph_cache.template get<tags::Output1>();
double func2_result = graph_cache.template get<tags::Output2>();

auto func1_status = report->template status<tags::Output1>();
auto func2_status = report->template status<tags::Output2>();

std::string_view func1_status_string = preparation_status_name_map[func1_status]; // note this is an inline map defined in
    serif::utils::dependencies
std::string_view func2_status_string = preparation_status_name_map[func2_status];

std::println("Func1 Result: {}, Func2 Result: {}, Func1 Status: {}, Func2 Status: {}", func1_result, func2_result,
    func1_status_string, func2_status_string);
```

Now what we could do is change only one part of the input context and call try_prepare again. Lets say we change only y, such that only func2 is stale.

```
y = 3.0;
auto report = graph_cache.try_prepare(graph_context);

double func1_result = graph_cache.template get<tags::Output1>();
double func2_result = graph_cache.template get<tags::Output2>();

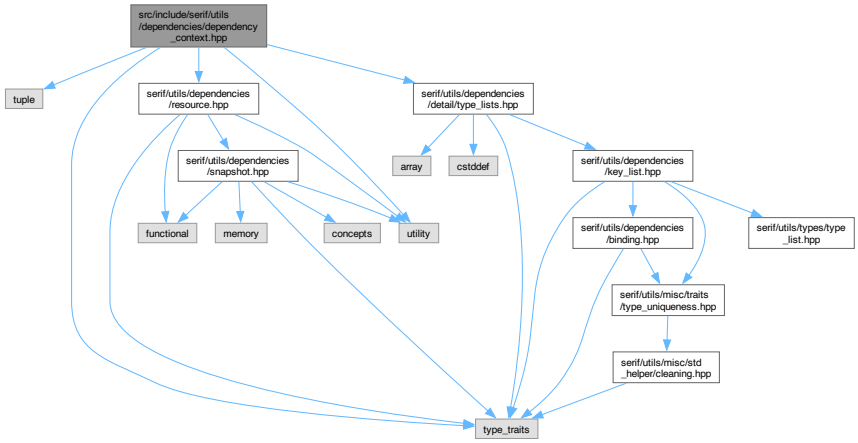
auto func1_status = report->template status<tags::Output1>();
auto func2_status = report->template status<tags::Output2>();

std::string_view func1_status_string = preparation_status_name_map[func1_status]; // note this is an inline map defined in
    serif::utils::dependencies
std::string_view func2_status_string = preparation_status_name_map[func2_status];

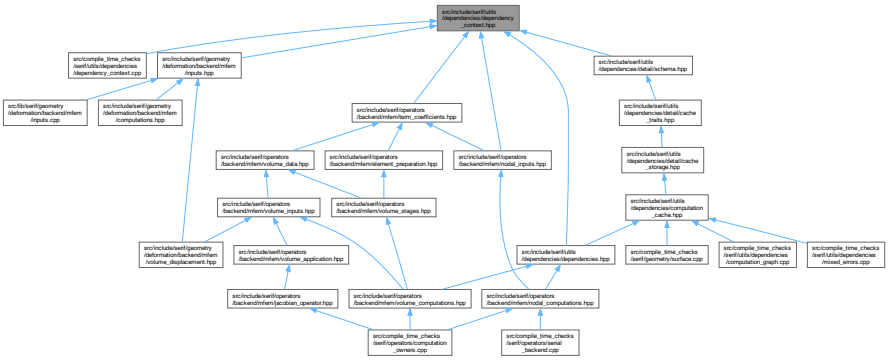
std::println("Func1 Result: {}, Func2 Result: {}, Func1 Status: {}, Func2 Status: {}", func1_result, func2_result,
    func1_status_string, func2_status_string);
```

11.203 src/include/serif/utils/dependencies/dependency_context.hpp File Reference

#include <tuple>
#include <type_traits>
#include <utility>
#include "serif/utils/dependencies/resource.hpp"
#include "serif/utils/dependencies/detail/type_lists.hpp"
Include dependency graph for dependency_context.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::utils::dependencies::DependencyContext< ResourceTs >
- class serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

Concepts

- concept serif::utils::dependencies::IsDependencyContext

Functions

- template<typename... Resources>
  requires (IsResource<Resources> && ...) && misc::traits::all_unique_types_v<typename std::remove_cvref_t<Resources>::Key...>
  auto serif::utils::dependencies::make_dependency_context (Resources &&... resources)

  Build a context whose schema follows directly from its resource keys and types.

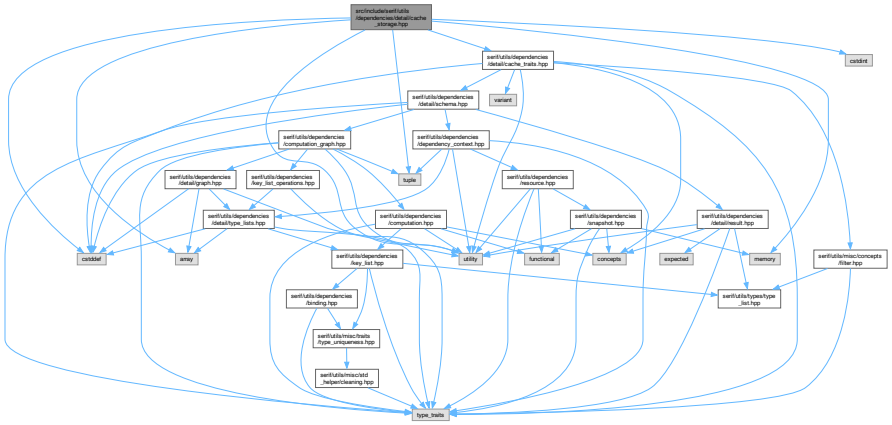
Variables

- template<typename T>
  constexpr bool serif::utils::dependencies::is_dependency_context_v = false
- template<typename... Resources>
  constexpr bool serif::utils::dependencies::is_dependency_context_v< DependencyContext< Resources... > >
  = true
- template<typename ParentT, typename KeysT>
  constexpr bool serif::utils::dependencies::is_dependency_context_v< DependencyContextView< ParentT, KeysT > >
  = true

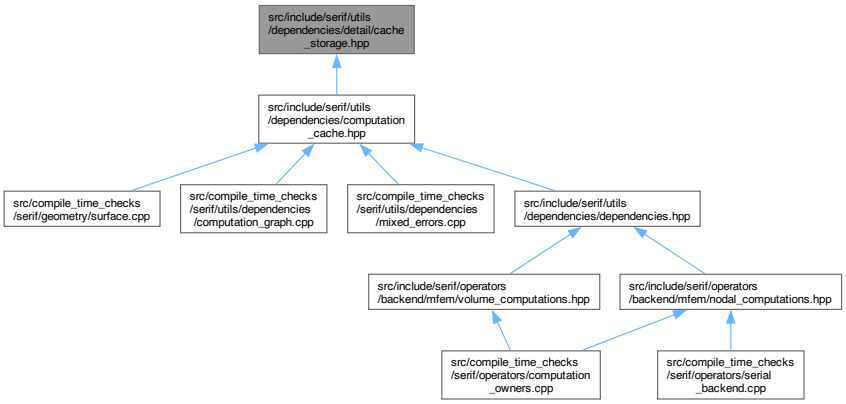
11.204 src/include/serif/utils/dependencies/detail/cache_storage.hpp File Reference

#include <array>
#include <cstddef>
#include <cstdint>
#include <memory>
#include <tuple>
#include <utility>

#include ”serif/utils/dependencies/detail/cache_traits.hpp”
Include dependency graph for cache_storage.hpp:



This graph shows which files directly or indirectly include this file:



Classes

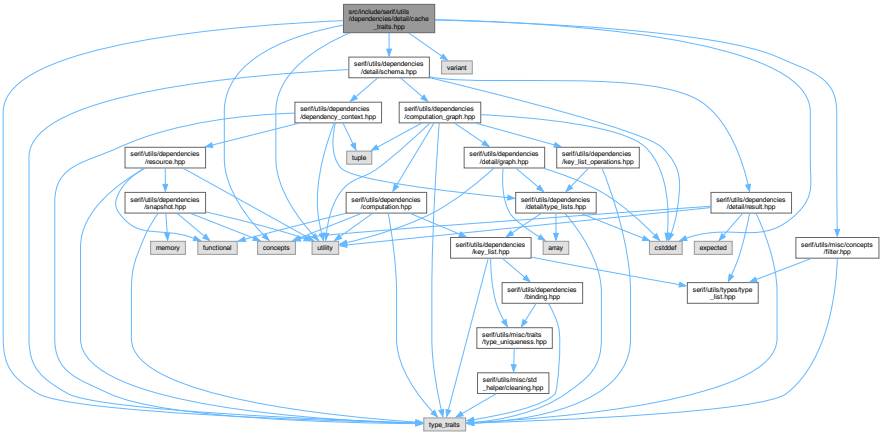
- struct serif::utils::dependencies::detail::SourceSnapshot< ResourceT >
- struct serif::utils::dependencies::detail::CachedResult< ValueT, InputCount >
- struct serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources..

Namespaces

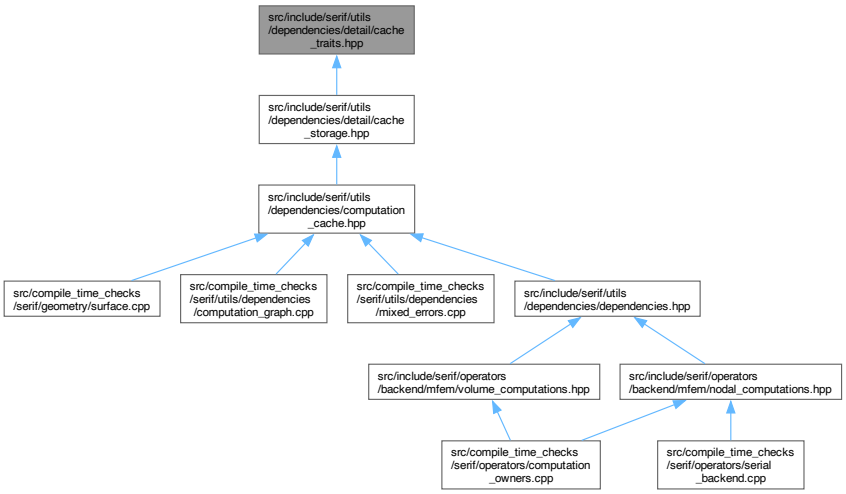
- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies
- namespace serif::utils::dependencies::detail

11.205 src/include/serif/utils/dependencies/detail/cache_traits.hpp File Reference

#include <concepts>
#include <cstddef>
#include <type_traits>
#include <utility>
#include <variant>
#include "serif/utils/dependencies/detail/schema.hpp"
#include "serif/utils/misc/concepts/filter.hpp"
Include dependency graph for cache_traits.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::detail::UsedResource< GraphT >
- struct serif::utils::dependencies::detail::UsedResource< GraphT >::Predicate< ResourceT >
- struct serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > >
- struct serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, Valid >
- struct serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies
- namespace serif::utils::dependencies::detail

Concepts

- concept serif::utils::dependencies::CanPrepare

Typedefs

- template<typename GraphT, typename ContextT>
  using serif::utils::dependencies::detail::UsedResources = misc::concepts::FilterT<typename ContextT↩
  ::Resources, UsedResource<GraphT>::template Predicate>

Functions

- template<typename GraphT, typename BindingsT, typename ErrorT, std::size_t... Indices>
  consteval bool serif::utils::dependencies::detail::compatible_errors (std::index_sequence< Indices... >)

  Value-returning calculations can share a graph with expected results carrying its declared error or an exact union alternative.

Variables

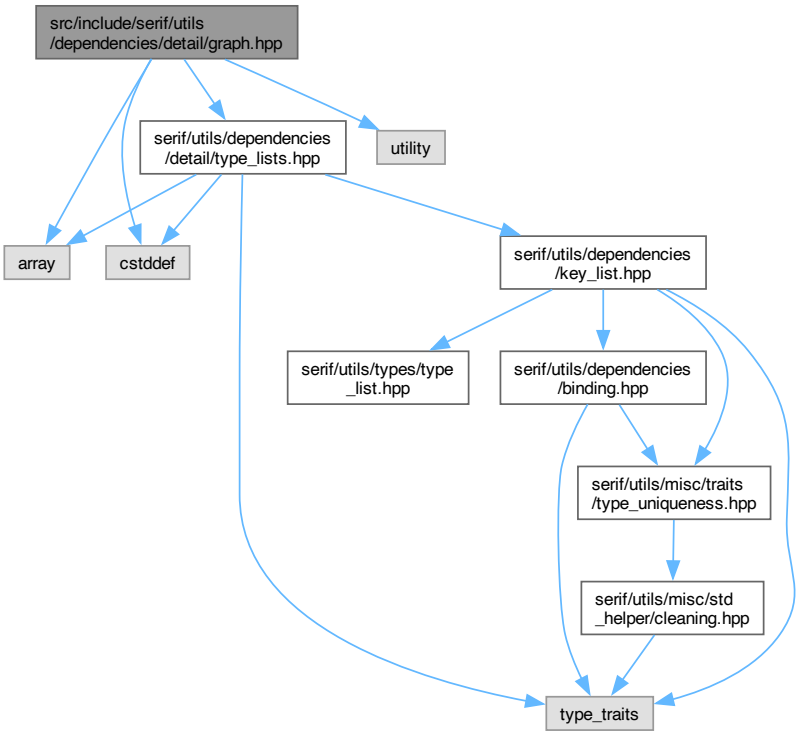
- template<typename Source, typename Target>
  constexpr bool serif::utils::dependencies::detail::compatible_error_v = std::same_as<Source, Target>

  Ordinary graph failures retain an exact type; numerical conversions must not erase diagnostics.
- template<typename Source, typename... Alternatives>
  constexpr bool serif::utils::dependencies::detail::compatible_error_v< Source, std::variant< Alternatives... > >

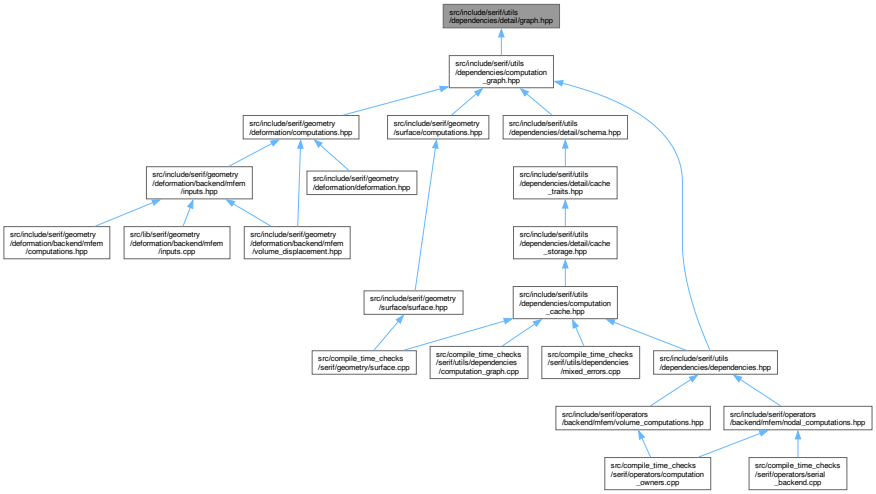
  Independent modules may explicitly aggregate their unchanged failures in an unambiguous tagged union.

11.206 src/include/serif/utils/dependencies/detail/graph.hpp File Reference

#include <array>
#include <cstddef>
#include <utility>
#include "serif/utils/dependencies/detail/type_lists.hpp"
Include dependency graph for graph.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::detail::Graph< Computations >

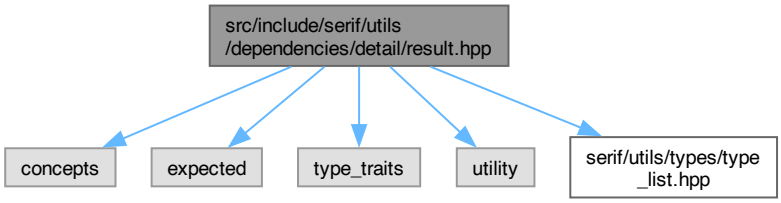
Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies
- namespace serif::utils::dependencies::detail

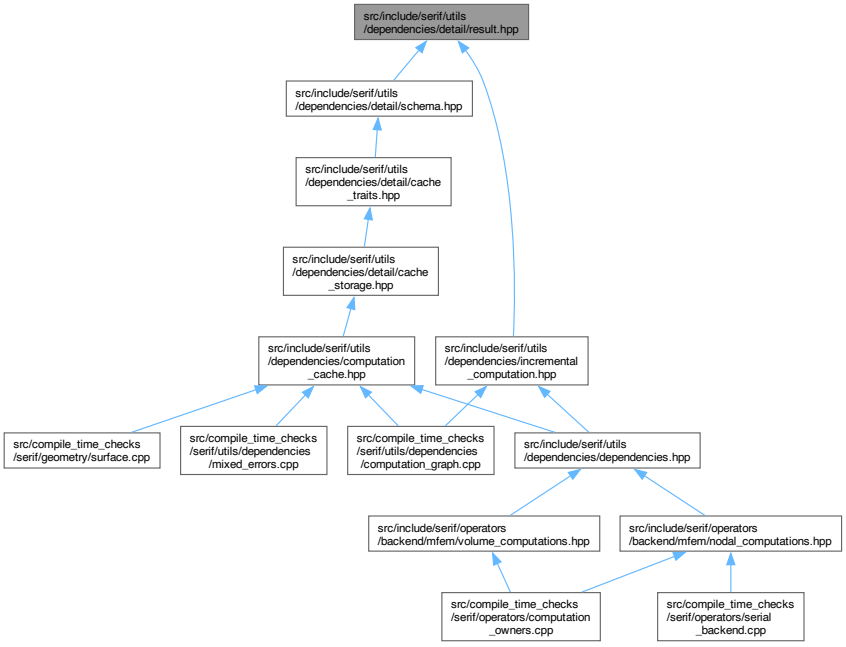
11.207 src/include/serif/utils/dependencies/detail/result.hpp File Reference

```
#include <concepts>
#include <expected>
#include <type_traits>
#include <utility>
```

#include "serif/utils/types/type_list.hpp"
Include dependency graph for result.hpp:



This graph shows which files directly or indirectly include this file:



Classes

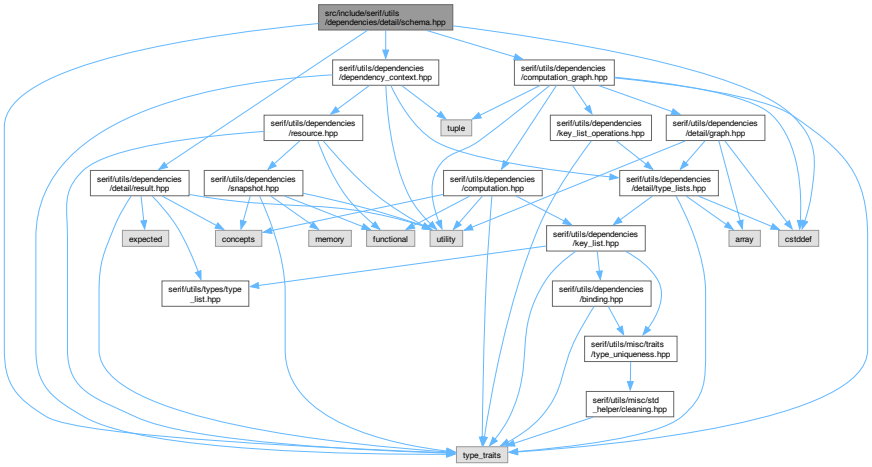
- struct serif::utils::dependencies::detail::Result< T >
- struct serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >
- struct serif::utils::dependencies::detail::Invocation< ComputationT, ArgumentsT, typename >
- struct serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< de

Namespaces

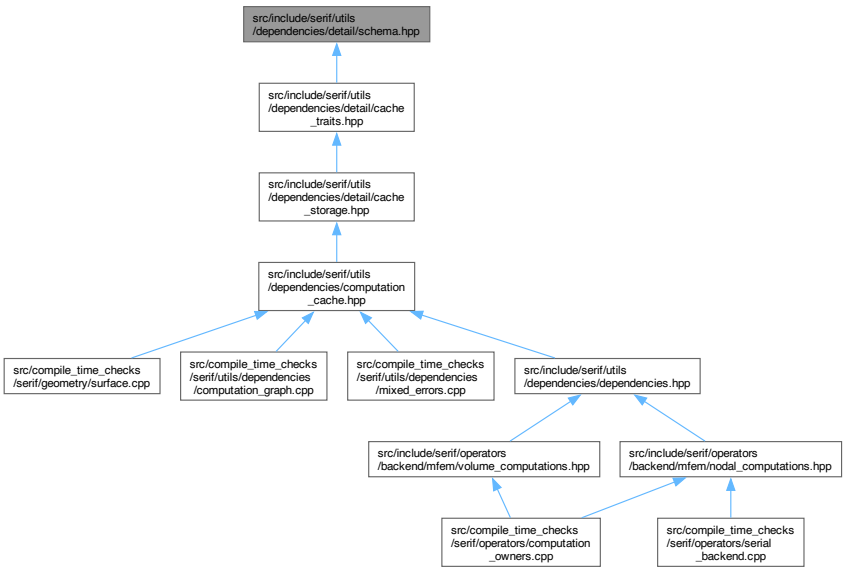
- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies
- namespace serif::utils::dependencies::detail

11.208 src/include/serif/utils/dependencies/detail/schema.hpp File Reference

#include <cstddef>
#include <type_traits>
#include "serif/utils/dependencies/dependency_context.hpp"
#include "serif/utils/dependencies/computation_graph.hpp"
#include "serif/utils/dependencies/detail/result.hpp"
Include dependency graph for schema.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::detail::AppendBinding< Bindings< BindingsTs... >, BindingT >
- struct serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, Available >
- struct serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, true >
- struct serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, Valid >
- struct serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, true >
- struct serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, false >
- struct serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true >
- struct serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, Independent >
- struct serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, true >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies
- namespace serif::utils::dependencies::detail

Concepts

- concept serif::utils::dependencies::ComputationGraphFor

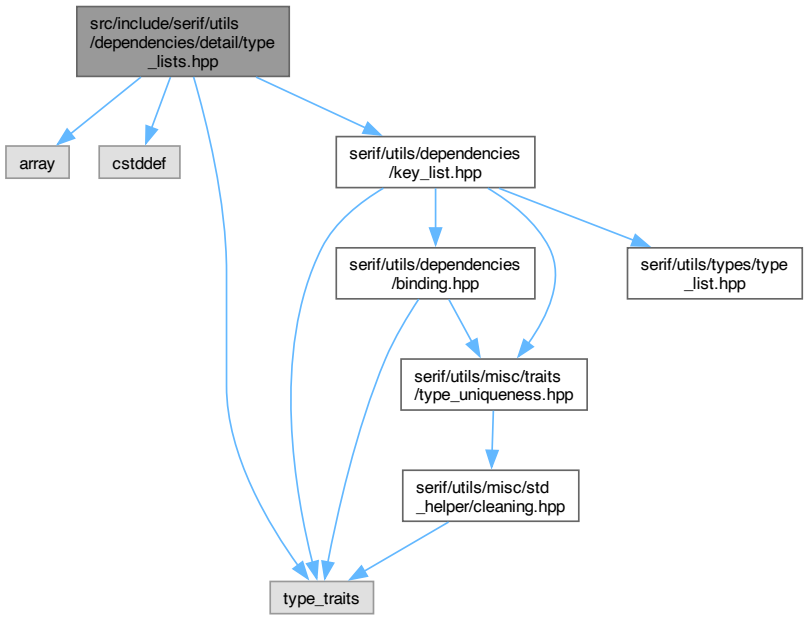
Functions

- template<typename BindingsT, typename... Keys>
  consteval bool serif::utils::dependencies::detail::independent_results (KeyList< Keys... >)

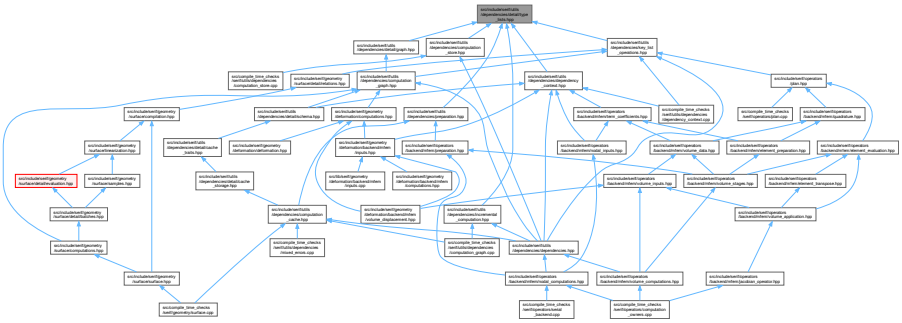
  A key has exactly one producer: either the context or a computation.

11.209 src/include/serif/utils/dependencies/detail/type_lists.hpp File Reference

#include <array>
#include <cstddef>
#include <type_traits>
#include "serif/utils/dependencies/key_list.hpp"
Include dependency graph for type_lists.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::detail::KeyListSize< KeyList< Keys... > >
- struct serif::utils::dependencies::detail::AppendUnique< KeyList< Keys... >, T >
- struct serif::utils::dependencies::detail::AppendKeys< ResultT, Keys >
- struct serif::utils::dependencies::detail::AppendKeys< ResultT, FirstT, Rest... >
- struct serif::utils::dependencies::detail::MergeKeyLists< ResultT, Lists >
- struct serif::utils::dependencies::detail::MergeKeyLists< ResultT, KeyList< Keys... >, Rest... >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies
- namespace serif::utils::dependencies::detail

Functions

- template<typename KeyT, typename... Types>
  consteval std::size_t serif::utils::dependencies::detail::type_index (types::TypeList< Types... >)

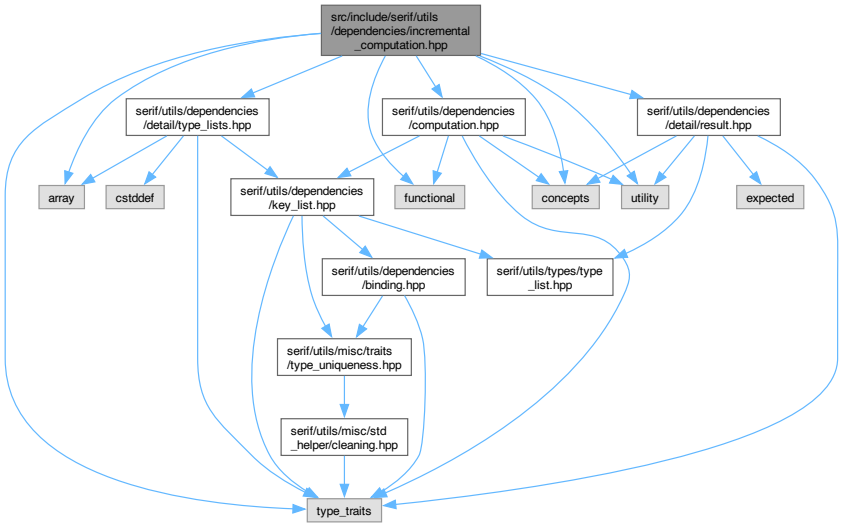
  Locate a type without requiring the types themselves to be constructible.

Variables

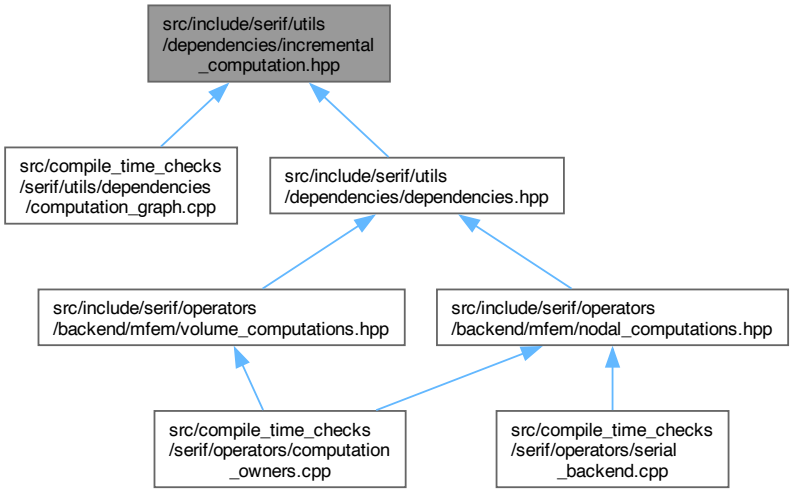
- template<typename KeyT, typename KeysT>
  constexpr bool serif::utils::dependencies::detail::contains_key_v = false
- template<typename KeyT, typename... Keys>
  constexpr bool serif::utils::dependencies::detail::contains_key_v< KeyT, KeyList< Keys... > > = (std::is_same_v<KeyT, Keys> || ...)
- template<typename RequiredT, typename AvailableT>
  constexpr bool serif::utils::dependencies::detail::key_list_subset_v = false
- template<typename... Keys, typename AvailableT>
  constexpr bool serif::utils::dependencies::detail::key_list_subset_v< KeyList< Keys... >, AvailableT > = (contains_key_v<Keys, AvailableT> && ...)

11.210 src/include/serif/utils/dependencies/incremental_computation.hpp File Reference

#include <array>
#include <concepts>
#include <functional>
#include <type_traits>
#include <utility>
#include "serif/utils/dependencies/computation.hpp"
#include "serif/utils/dependencies/detail/result.hpp"
#include "serif/utils/dependencies/detail/type_lists.hpp"
Include dependency graph for incremental_computation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >
- class serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >

  A calculation may reuse unchanged pieces of its previous immutable result.

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

Functions

- template<typename KeyT, typename ValueT, IsKeyList InputsT, typename FunctionT>
  requires std::is_object_v<ValueT> && std::move_constructible<ValueT> && std::constructible_from<std::decay_t<FunctionT>, FunctionT>
  auto serif::utils::dependencies::make_incremental_computation (FunctionT &&function)

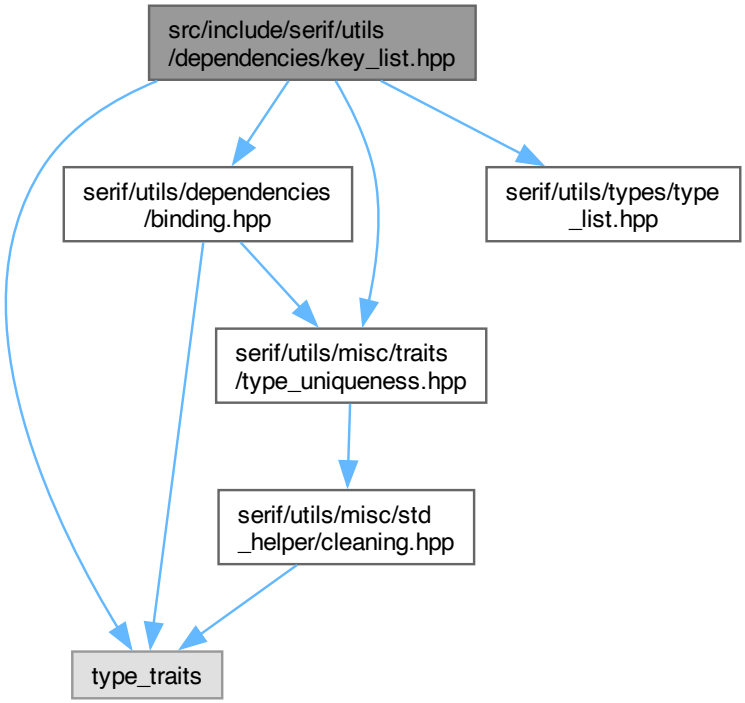
  Declare the result type explicitly so an incremental callable can inspect its previous value.

Variables

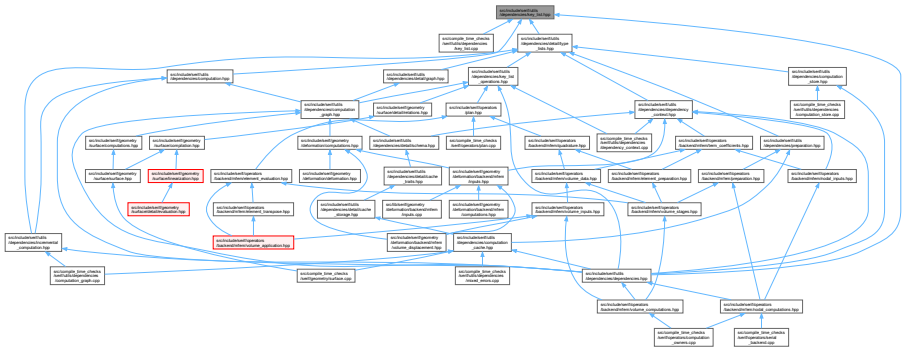
- template<typename KeyT, typename ValueT, typename InputsT, typename FunctionT>
  constexpr bool serif::utils::dependencies::is_computation_v< IncrementalComputation< KeyT, ValueT, InputsT, Func
  = true

11.211 src/include/serif/utils/dependencies/key_list.hpp File Reference

#include <type_traits>
#include "serif/utils/dependencies/binding.hpp"
#include "serif/utils/misc/traits/type_uniqueness.hpp"
#include "serif/utils/types/type_list.hpp"
Include dependency graph for key_list.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::KeyList< KeyTs >

  An ordered collection of role keys, independent of their associated value types.
- struct serif::utils::dependencies::KeyListBindings< BindingsT, KeyList< KeyTs... > >
- struct serif::utils::dependencies::ResolveKeys< BindingsT, KeyList< KeyTs... > >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

Concepts

- concept serif::utils::dependencies::IsKeyList
- concept serif::utils::dependencies::SatisfiesKeyList

Typedefs

- template<typename... KeyTs>
  using serif::utils::dependencies::InputList = KeyList<KeyTs...>

  Computation arguments are supplied in this key order.
- template<typename... KeyTs>
  using serif::utils::dependencies::RequestedResults = KeyList<KeyTs...>

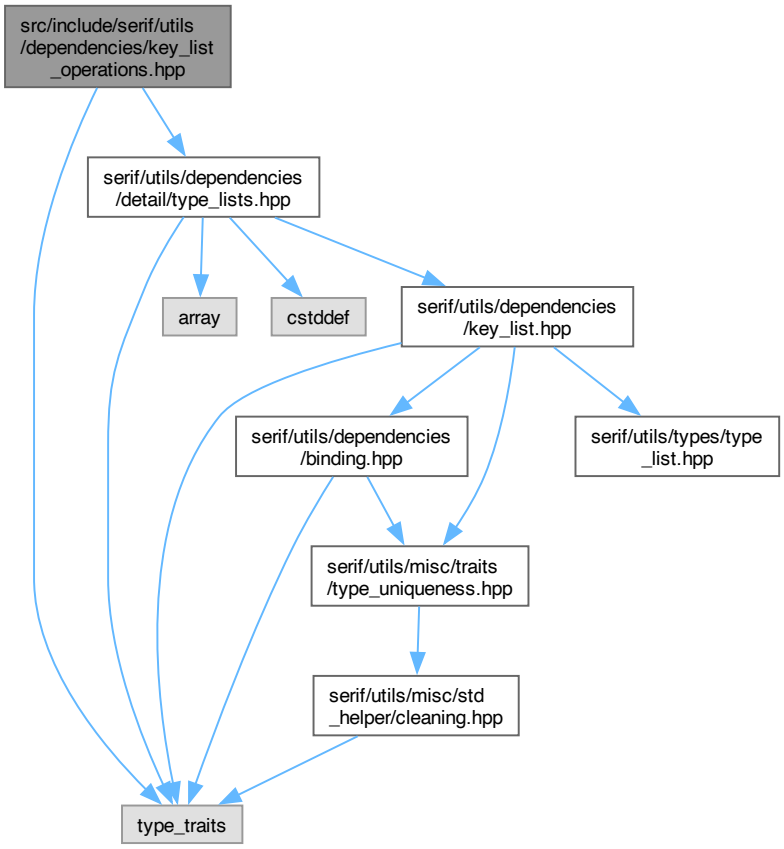
  Preparing these results also prepares their required intermediate computations.
- template<typename BindingsT, typename KeysT>
  using serif::utils::dependencies::ResolvedTargets = ResolveKeys<BindingsT, std::remove_cvref_t<KeysT>>::type

Variables

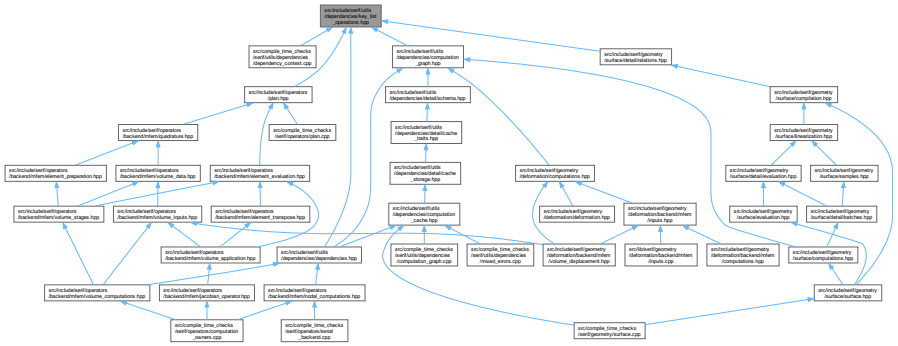
- template<typename T>
  constexpr bool serif::utils::dependencies::is_key_list_v = false
- template<typename... KeyTs>
  constexpr bool serif::utils::dependencies::is_key_list_v< KeyList< KeyTs... > > = misc::traits::all_unique_types_v<KeyTs...>

11.212 src/include/serif/utils/dependencies/key_list_operations.hpp File Reference

#include <type_traits>
#include "serif/utils/dependencies/detail/type_lists.hpp"
Include dependency graph for key_list_operations.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

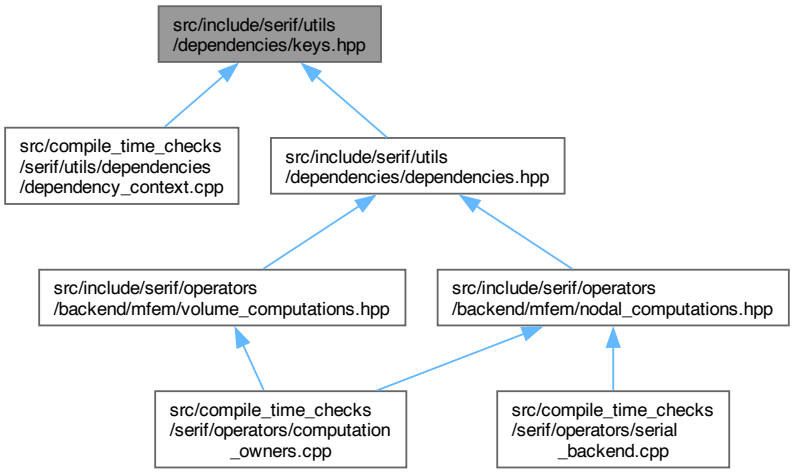
Typedefs

- template<IsKeyList... Lists>
  using serif::utils::dependencies::CombinedKeyList = detail::MergeKeyLists<KeyList<>, std::remove_cvref_t<Lists>...>::type

  Combine requirements in first-appearance order, retaining distinct keys with shared targets.

11.213 src/include/serif/utils/dependencies/keys.hpp File Reference

This graph shows which files directly or indirectly include this file:



Classes

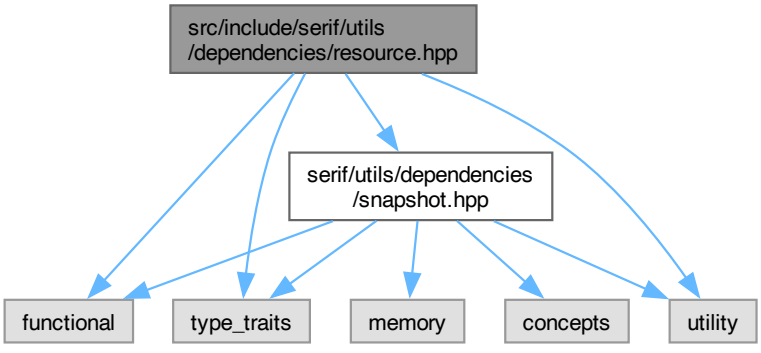
- struct serif::utils::dependencies::keys::Input
  The main input to a computation.
- struct serif::utils::dependencies::keys::Output
  The main result of a computation.
- struct serif::utils::dependencies::keys::Scale
  A multiplicative scale supplied as an input.
- struct serif::utils::dependencies::keys::Parameters
  Configuration values that can affect the result.
- struct serif::utils::dependencies::keys::Coordinates
  Positions supplied to a computation, in the caller's coordinate convention.

Namespaces

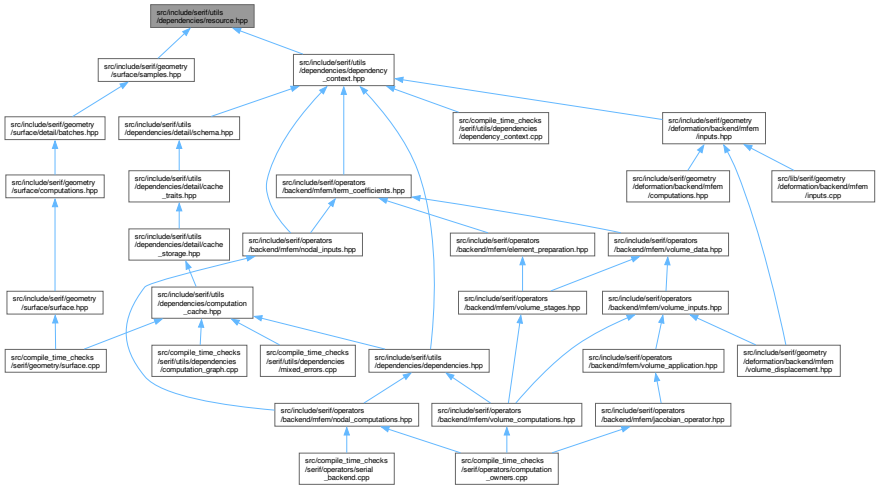
- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies
- namespace serif::utils::dependencies::keys

11.214 src/include/serif/utils/dependencies/resource.hpp File Reference

#include <functional>
#include <type_traits>
#include <utility>
#include "serif/utils/dependencies/snapshot.hpp"
Include dependency graph for resource.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::detail::OwnedValue< T >
- struct serif::utils::dependencies::detail::BorrowedValue< T >
- class serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies
- namespace serif::utils::dependencies::detail

Concepts

- concept serif::utils::dependencies::IsResource

Functions

- template<typename KeyT, typename ValueT, typename PolicyT = ValueSnapshot>
  auto serif::utils::dependencies::own (ValueT &&value, PolicyT policy={})

  Store a value in the context; rvalues transfer ownership and lvalues are copied.
- template<typename KeyT, typename ValueT, typename PolicyT = ValueSnapshot>
  auto serif::utils::dependencies::borrow (ValueT &value, PolicyT policy={})

  Bind an existing object whose lifetime must cover every use of the context.
- template<typename KeyT, typename ValueT, typename PolicyT = ValueSnapshot>
  auto serif::utils::dependencies::borrow (const ValueT &&, PolicyT={})=delete

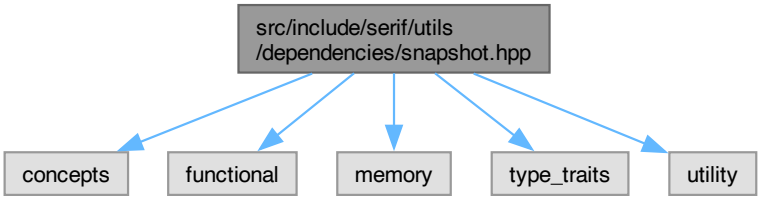
  Borrowed storage must never refer to a temporary, including a const temporary.

Variables

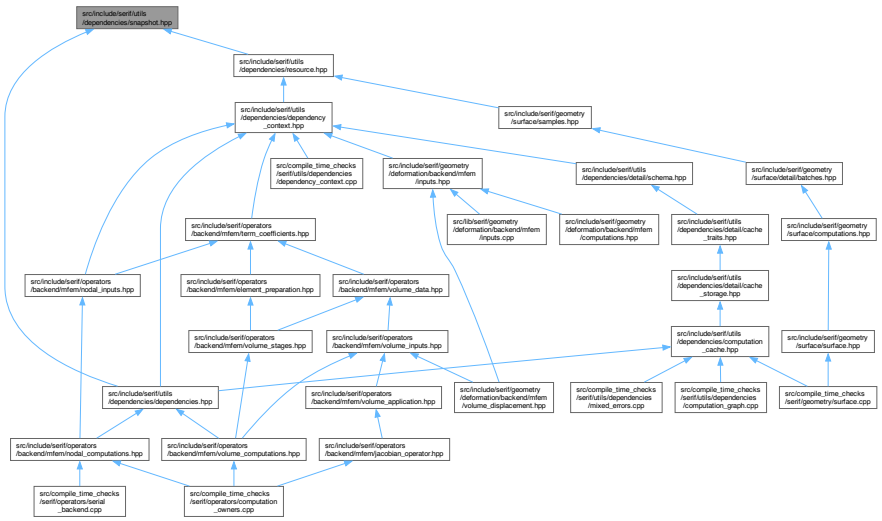
- template<typename T>
  constexpr bool serif::utils::dependencies::is_resource_v = false
- template<typename KeyT, typename StorageT, typename PolicyT>
  constexpr bool serif::utils::dependencies::is_resource_v< Resource< KeyT, StorageT, PolicyT > > = true

11.215 src/include/serif/utils/dependencies/snapshot.hpp File Reference

#include <concepts>
#include <functional>
#include <memory>
#include <type_traits>
#include <utility>
Include dependency graph for snapshot.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::dependencies::ValueSnapshot

  Default tracking owns an exact value copy; noncopyable resources supply another policy.
- class serif::utils::dependencies::ProjectedSnapshot< FunctionT >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::dependencies

Concepts

- concept serif::utils::dependencies::SnapshotPolicyFor

Functions

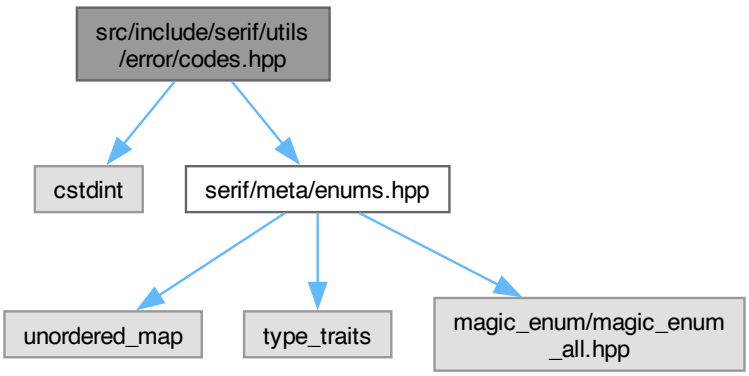
- template<typename FunctionT>
  auto serif::utils::dependencies::track_by (FunctionT &&function)

  Select the state that determines whether a source needs recomputation.
- template<typename PolicyT, typename ValueT>
  requires SnapshotPolicyFor<PolicyT, ValueT>
  const void * serif::utils::dependencies::source_identity (const PolicyT &policy, const ValueT &value)

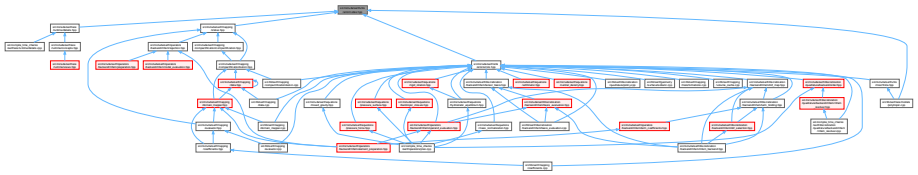
  Views may identify their underlying resource; ordinary sources retain object identity.

11.216 src/include/serif/utils/error/codes.hpp File Reference

#include <cstdint>
#include "serif/meta/enums.hpp"
Include dependency graph for codes.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::errors

Enumerations

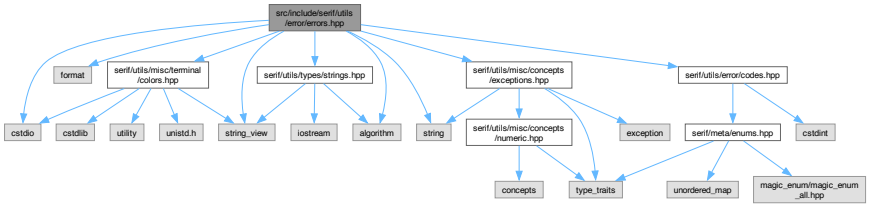
- enum class serif::utils::errors::SERiFErrorCode : uint16_t {
  serif::utils::errors::UNKNOWN_ERROR = 0 , serif::utils::errors::INVALID_ARGUMENT = 100 ,
  serif::utils::errors::INVALID_QUADRATURE_ORDER , serif::utils::errors::INVALID_DOMAIN_ERROR
  =110 ,
  serif::utils::errors::NEGATIVE_VALUE_ERROR , serif::utils::errors::NON_FINITE_VALUE_ERROR
  , serif::utils::errors::JACOBIAN_ERROR = 200 , serif::utils::errors::INVERTED_JACOBIAN ,
  serif::utils::errors::NON_FINITE_DETERMINATE , serif::utils::errors::EOS_ERROR = 300 ,
  serif::utils::errors::EOS_UNSUPPORTED_RELATION , serif::utils::errors::EOS_UNSUPPORTED_DERIVATIVE
  ,
  serif::utils::errors::EOS_WRONG_INPUT_COUNT , serif::utils::errors::EOS_WRONG_INPUT_QUANTITY
  , serif::utils::errors::EOS_NONFINITE_INPUT , serif::utils::errors::EOS_OUTSIDE_DOMAIN ,
  serif::utils::errors::EOS_NONFINITE_RESULT , serif::utils::errors::EOS_INVALID_THERMODYNAMIC_INPUT
  , serif::utils::errors::INVALID_DIMENSION = 400 , serif::utils::errors::NON_FINITE_MAPPING_INPUT
  ,
  serif::utils::errors::INVALID_REFERENCE_RADIUS , serif::utils::errors::AT_COMPACTIFIED_INFINITY
  , serif::utils::errors::OUTSIDE_REFERENCE_DOMAIN , serif::utils::errors::NON_FINITE_MAPPING_RESULT
  ,
  serif::utils::errors::NON_POSITIVE_DETERMINATE , serif::utils::errors::INVALID_MAPPING ,
  serif::utils::errors::MISC_MAPPING_ERROR =410 , serif::utils::errors::INVALID_DOF_COUNT ,
  serif::utils::errors::INVALID_ELEMENT_GEOMETRY , serif::utils::errors::INVALID_DOF_ORDERING
  , serif::utils::errors::INVALID_COMPACTIFICATION_RANK , serif::utils::errors::INVALID_COMPACTIFICATION
  ,
  serif::utils::errors::INVALID_COMPACTIFICATION_DERIVATIVE_TYPE , serif::utils::errors::INVALID_VACUUM
  , serif::utils::errors::MISSING_EXTERIOR_MAP , serif::utils::errors::COMPACTIFICATION_MISMATCH
  ,
  serif::utils::errors::MISSING_FACE_ELEMENT , serif::utils::errors::JACOBIAN_SHAPE_ERROR ,
  serif::utils::errors::MISSING_DISPLACEMENT_FINITE_ELEMENT_SPACE , serif::utils::errors::MISMATCHED_
  ,
  serif::utils::errors::IMPROPERLY_ORDERED_REFERENCE_RADII = 450 , serif::utils::errors::INVALID_COORDI
  , serif::utils::errors::EVALUATOR_ERROR =460 , serif::utils::errors::CACHE_FE_BINDING_ERROR
  ,
  serif::utils::errors::INVALID_ELEMENT_ID_ERROR , serif::utils::errors::ELEMENT_NULLPTR_ERROR
  , serif::utils::errors::MATRIX_SHAPE_ERROR = 500 , serif::utils::errors::OUT_OF_RANGE_ERROR
  ,
  serif::utils::errors::FAILED_TYPE_INVARIANT = 1000 }

Variables

- auto serif::utils::errors::SERiFErrorCodeNameToTypeMap = meta::enum_to_string_enum_map<SERiFErrorCode>(
- auto serif::utils::errors::SERiFErrorCodeTypeToNameMap = meta::enum_to_enum_string_map<SERiFErrorCode>(

11.217 src/include/serif/utils/error/errors.hpp File Reference

#include <string_view>
#include <format>
#include <cstdio>
#include <string>
#include <algorithm>
#include "serif/utils/misc/terminal/colors.hpp"
#include "serif/utils/misc/concepts/exceptions.hpp"
#include "serif/utils/error/codes.hpp"
#include "serif/utils/types/strings.hpp"
Include dependency graph for errors.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

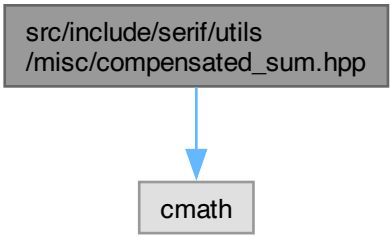
- namespace serif
- namespace serif::utils
- namespace serif::utils::errors

Functions

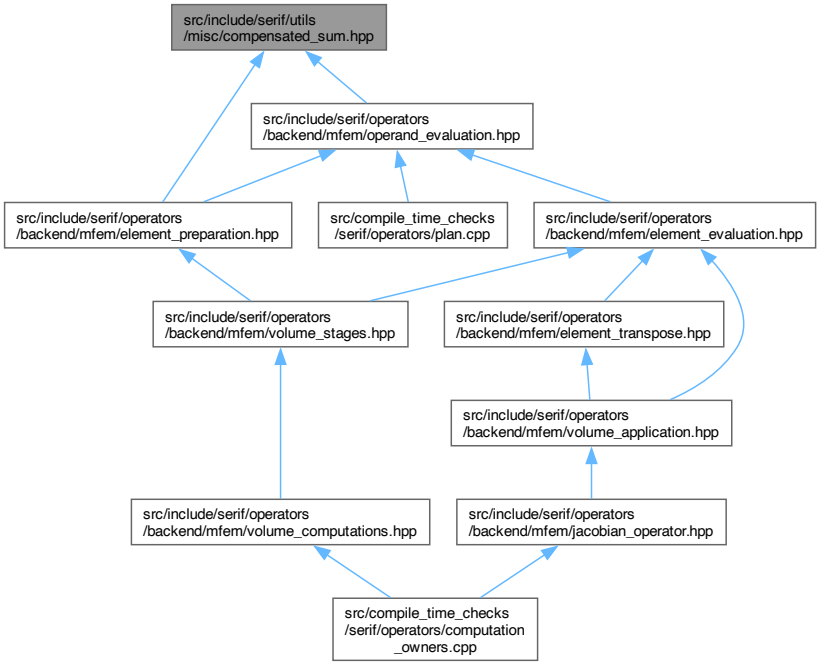
- template<misc::concepts::IsException ExceptionT, SERiFErrorCode code>
  void serif::utils::errors::serif_error (const std::string_view message, auto... args)
- template<SERiFErrorCode code, misc::concepts::IsException ExceptionT = std::logic_error>
  void serif::utils::errors::serif_assert (const bool condition, const std::string_view message, auto... args)
- template<SERiFErrorCode code, types::ct_error_string message, bool condition>
  constexpr void serif::utils::errors::serif_assert ()
- template<SERiFErrorCode code, misc::concepts::IsException ExceptionT = std::logic_error>
  void serif::utils::errors::serif_ensure (const bool condition, const std::string_view message, auto... args)
- template<SERiFErrorCode code, types::ct_error_string message, bool condition>
  constexpr void serif::utils::errors::serif_ensure ()

11.218 src/include/serif/utils/misc/compensated_sum.hpp File Reference

#include <cmath>
Include dependency graph for compensated_sum.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class serif::utils::CompensatedSum

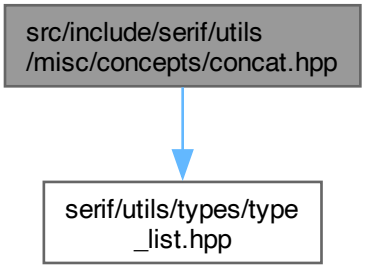
  Neumaier summation retains small terms when a scalar reduction contains cancellation.

Namespaces

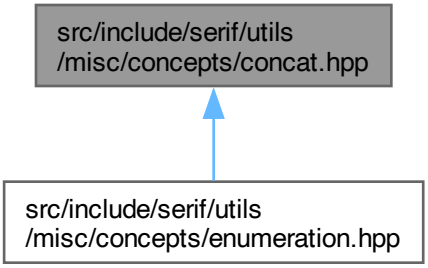
- namespace serif
- namespace serif::utils

11.219 src/include/serif/utils/misc/concepts/concat.hpp File Reference

#include "serif/utils/types/type_list.hpp"
Include dependency graph for concat.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::misc::concepts::Concat<>
- struct serif::utils::misc::concepts::Concat< types::TypeList< Ts... > >
- struct serif::utils::misc::concepts::Concat< types::TypeList< LhsTs... >, types::TypeList< RhsTs... >, Rest... >

Namespaces

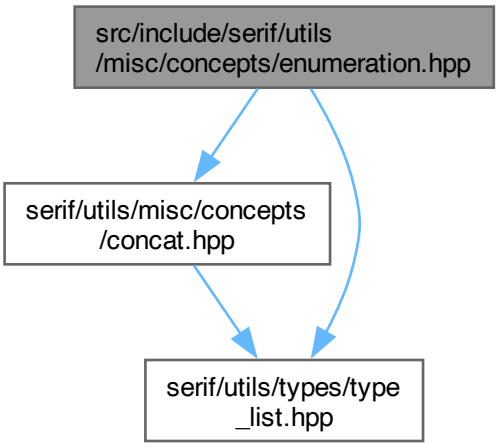
- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::concepts

Typedefs

- template<typename... Lists>
  using serif::utils::misc::concepts::ConcatT = Concat<Lists...>::Type

11.220 src/include/serif/utils/misc/concepts/enumeration.hpp File Reference

#include "serif/utils/misc/concepts/concat.hpp"
#include "serif/utils/types/type_list.hpp"
Include dependency graph for enumeration.hpp:



Classes

- struct serif::utils::misc::concepts::EnumerateTypes< types::TypeList< TypeTs... >, Descriptor >

Namespaces

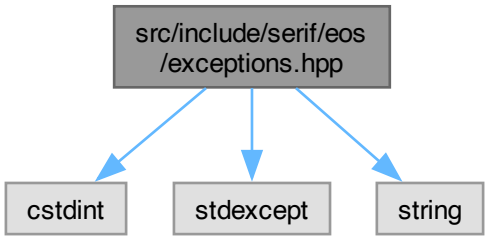
- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::concepts

Typedefs

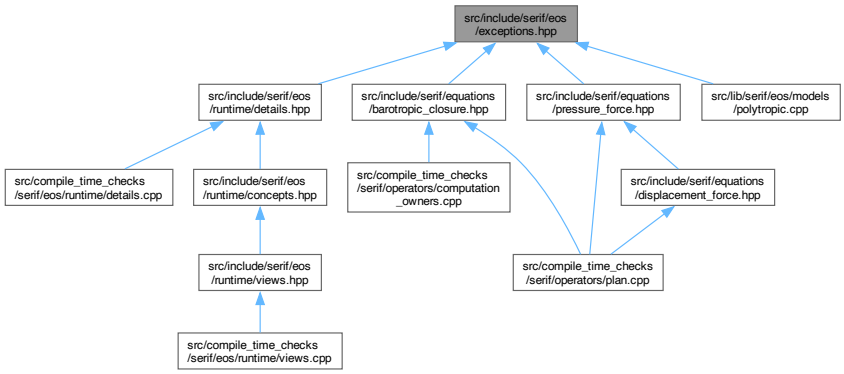
- template<typename TypeList, template< typename > typename Descriptor>
  using serif::utils::misc::concepts::EnumerateTypesT = EnumerateTypes<TypeList, Descriptor>::Type

11.221 src/include/serif/eos/exceptions.hpp File Reference

#include <cstdint>
#include <stdexcept>
#include <string>
Include dependency graph for exceptions.hpp:



This graph shows which files directly or indirectly include this file:



Classes

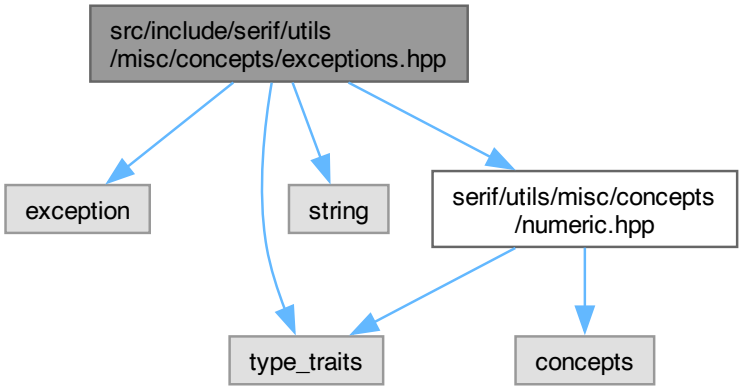
- class serif::eos::EOSEvaluationError

Namespaces

- namespace serif
- namespace serif::eos

11.222 src/include/serif/utils/misc/concepts/exceptions.hpp File Reference

#include <exception>
#include <type_traits>
#include <string>
#include "serif/utils/misc/concepts/numeric.hpp"
Include dependency graph for exceptions.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

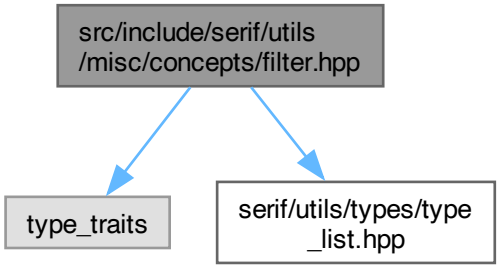
- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::concepts

Concepts

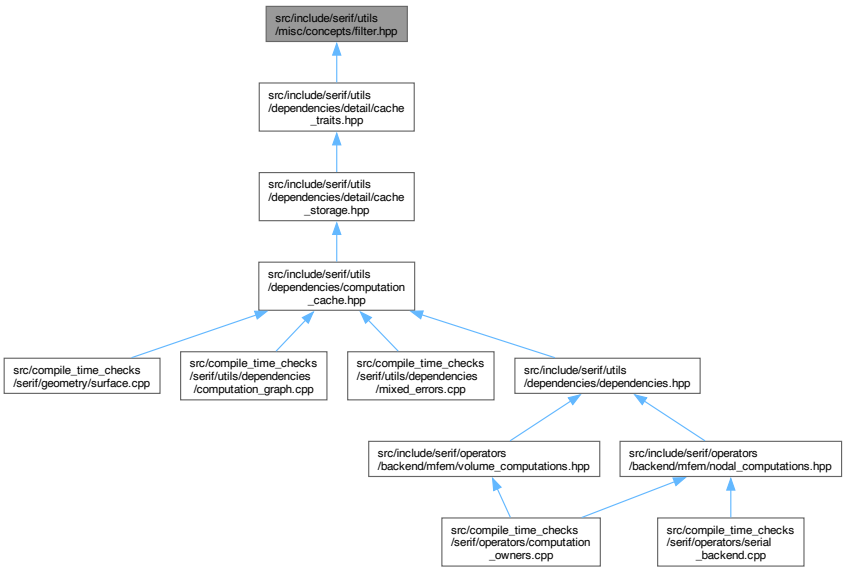
- concept serif::utils::misc::concepts::IsException
- concept serif::utils::misc::concepts::IsExceptionWithErrorCode

11.223 src/include/serif/utils/misc/concepts/filter.hpp File Reference

#include <type_traits>
#include "serif/utils/types/type_list.hpp"
Include dependency graph for filter.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::misc::concepts::Prepend< T, types::TypeList< Ts... > >
- struct serif::utils::misc::concepts::Filter< types::TypeList<>, Predicate >
- struct serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate >

Namespaces

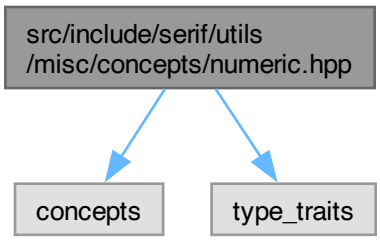
- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::concepts

Typedefs

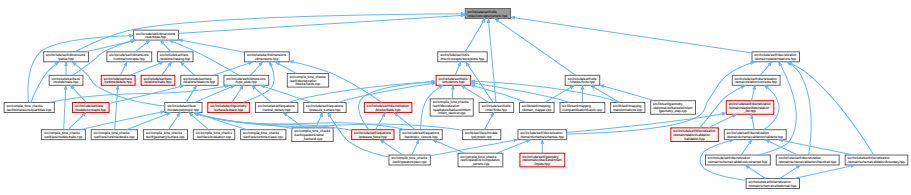
- template<typename T, typename List>
  using serif::utils::misc::concepts::PrependT = Prepend<T, List>::Type
- template<typename List, template< typename > typename Predicate>
  using serif::utils::misc::concepts::FilterT = Filter<List, Predicate>::Type

11.224 src/include/serif/utils/misc/concepts/numeric.hpp File Reference

#include <concepts>
#include <type_traits>
Include dependency graph for numeric.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

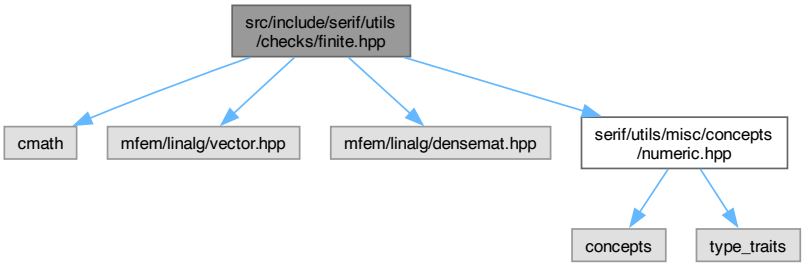
- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::concepts

Concepts

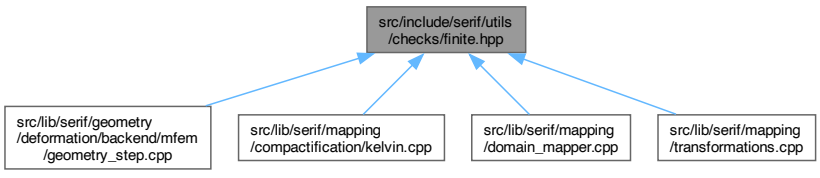
- concept serif::utils::misc::concepts::IsNumeric
- concept serif::utils::misc::concepts::IsOrderable
- concept serif::utils::misc::concepts::IsComparableToNumeric
- concept serif::utils::misc::concepts::SetOfOneOrTwo
- concept serif::utils::misc::concepts::IsEnum

11.225 src/include/serif/utils/checks/finite.hpp File Reference

#include <cmath>
#include "mfem/linalg/vector.hpp"
#include "mfem/linalg/densemat.hpp"
#include "serif/utils/misc/concepts/numeric.hpp"
Include dependency graph for finite.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

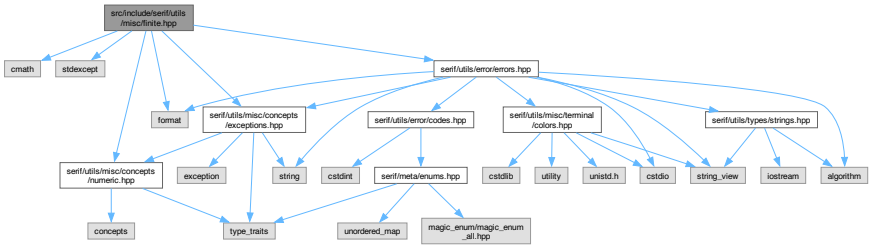
- namespace serif
- namespace serif::utils
- namespace serif::utils::checks

Functions

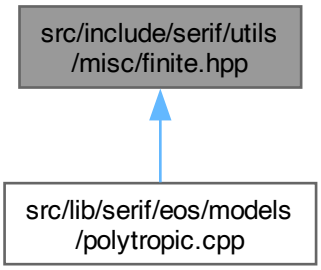
- bool serif::utils::checks::is_finite (const mfem::Vector &vector)
- bool serif::utils::checks::is_finite (const mfem::DenseMatrix &matrix)
- template<typename Scalar>
  requires requires (Scalar x) {{std::isfinite(x)} -> std::convertible_to<bool>; }
  bool serif::utils::checks::is_finite (const Scalar scalar)

11.226 src/include/serif/utils/misc/finite.hpp File Reference

#include <cmath>
#include <stdexcept>
#include <format>
#include "serif/utils/misc/concepts/numeric.hpp"
#include "serif/utils/misc/concepts/exceptions.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for finite.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

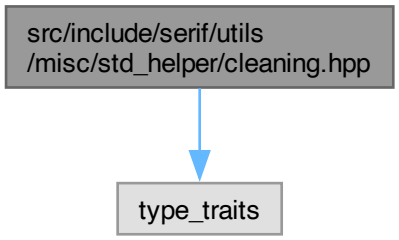
- namespace serif
- namespace serif::utils
- namespace serif::utils::misc

Functions

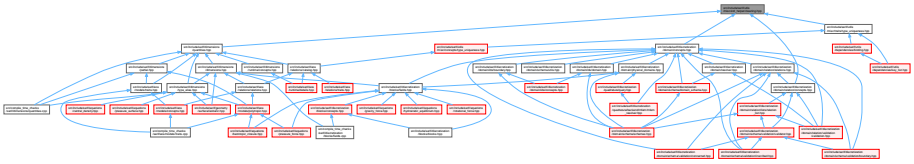
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsNumeric Scalar>
  void serif::utils::misc::validate_finite (const Scalar value)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void serif::utils::misc::validate_greater_than (const Scalar value, const Scalar predicate)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void serif::utils::misc::validate_less_than (const Scalar value, const Scalar predicate)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void serif::utils::misc::validate_nonnegativity (const Scalar value)

11.227 src/include/serif/utils/misc/std_helper/cleaning.hpp File Reference

#include <type_traits>
Include dependency graph for cleaning.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::std_helper

Typedefs

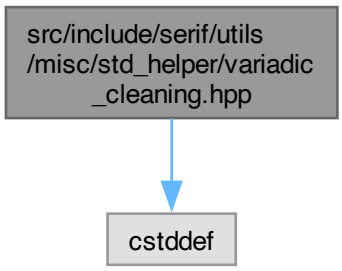
- template<class T>
  using serif::utils::misc::std_helper::NeitherConstNorVolatile = std::remove_cv_t<T>
- template<class T>
  using serif::utils::misc::std_helper::NeitherConstNorVolatileReference = std::remove_cvref_t<T>

Variables

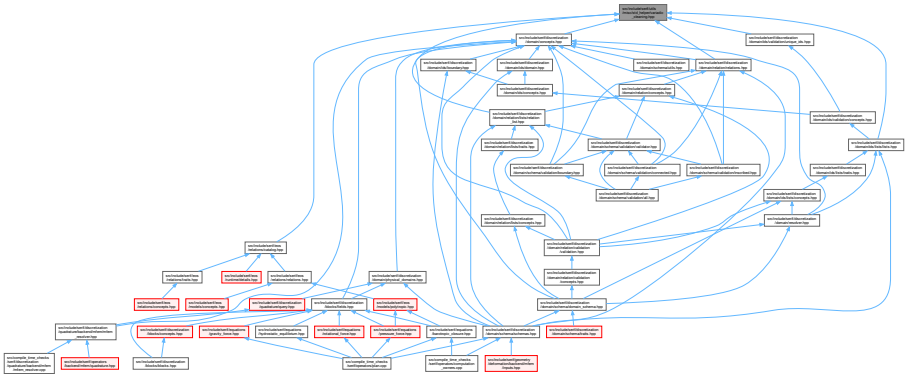
- template<class Child, class Parent>
  constexpr bool serif::utils::misc::std_helper::AIsBaseClassOfB = std::is_base_of_v<Parent, Child>
- template<class Ta, class Tb>
  constexpr bool serif::utils::misc::std_helper::SameType = std::is_same_v<Ta, Tb>

11.228 src/include/serif/utils/misc/std_helper/variadic_cleaning.hpp File Reference

#include <cstddef>
Include dependency graph for variadic_cleaning.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

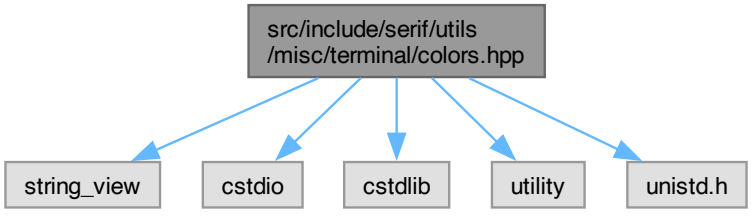
- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::std_helper

Functions

- template<typename... Ts>
  consteval std::size_t serif::utils::misc::std_helper::CountVariadicArguments () noexcept

11.229 src/include/serif/utils/misc/terminal/colors.hpp File Reference

#include <string_view>
#include <cstdio>
#include <cstdlib>
#include <utility>
#include <unistd.h>
Include dependency graph for colors.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::terminal

Enumerations

- enum class serif::utils::misc::terminal::ANSIColors : int { serif::utils::misc::terminal::BLACK = 30 , serif::utils::misc::terminal::RED = 31 , serif::utils::misc::terminal::GREEN = 32 , serif::utils::misc::terminal::YELLOW = 33 , serif::utils::misc::terminal::BLUE = 34 , serif::utils::misc::terminal::MAGENTA = 35 , serif::utils::misc::terminal::CYAN = 36 , serif::utils::misc::terminal::WHITE = 37 , serif::utils::misc::terminal::BRIGHT_BLACK = 90 , serif::utils::misc::terminal::BRIGHT_RED = 91 , serif::utils::misc::terminal::BRIGHT_GREEN = 92 , serif::utils::misc::terminal::RESET = 0 }

Functions

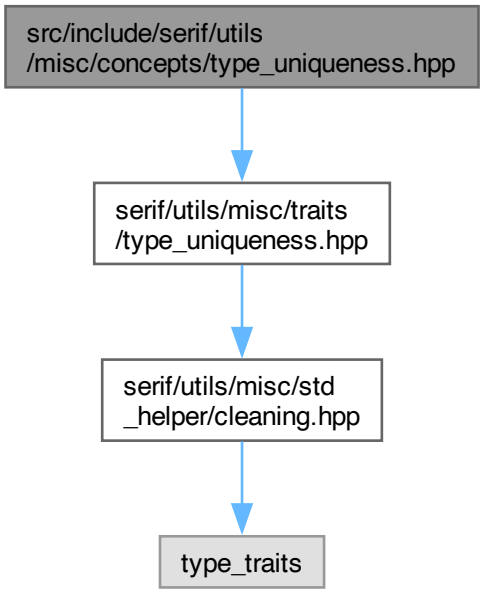
- consteval std::string_view serif::utils::misc::terminal::resolve_ansi_color (const ANSIColors color) noexcept
- bool serif::utils::misc::terminal::supports_ansi (std::FILE *stream=stdout) noexcept

Variables

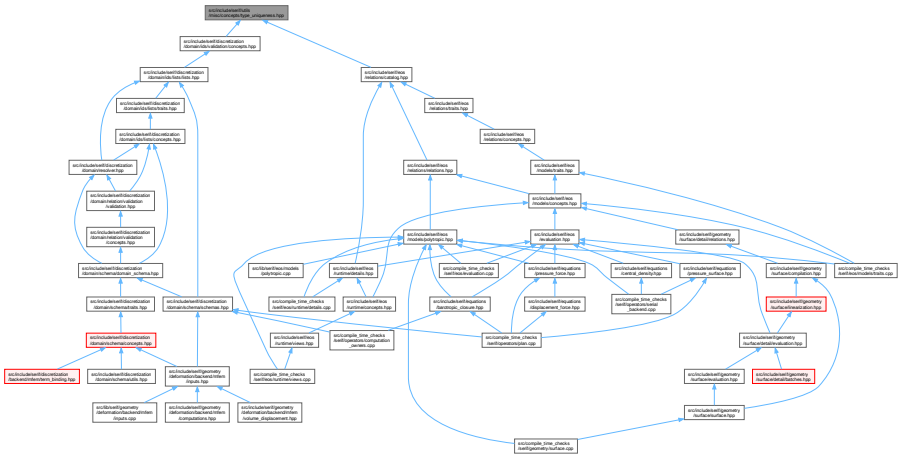
- constexpr std::string_view serif::utils::misc::terminal::a_black = resolve_ansi_color(ANSIColors::BLACK)
- constexpr std::string_view serif::utils::misc::terminal::a_red = resolve_ansi_color(ANSIColors::RED)
- constexpr std::string_view serif::utils::misc::terminal::a_green = resolve_ansi_color(ANSIColors::GREEN)
- constexpr std::string_view serif::utils::misc::terminal::a_yellow = resolve_ansi_color(ANSIColors::YELLOW)
- constexpr std::string_view serif::utils::misc::terminal::a_blue = resolve_ansi_color(ANSIColors::BLUE)
- constexpr std::string_view serif::utils::misc::terminal::a_magenta = resolve_ansi_color(ANSIColors::MAGENTA)
- constexpr std::string_view serif::utils::misc::terminal::a_cyan = resolve_ansi_color(ANSIColors::CYAN)
- constexpr std::string_view serif::utils::misc::terminal::a_white = resolve_ansi_color(ANSIColors::WHITE)
- constexpr std::string_view serif::utils::misc::terminal::a_bright_black = resolve_ansi_color(ANSIColors::BRIGHT_BL
- constexpr std::string_view serif::utils::misc::terminal::a_bright_red = resolve_ansi_color(ANSIColors::BRIGHT_RED
- constexpr std::string_view serif::utils::misc::terminal::a_bright_green = resolve_ansi_color(ANSIColors::BRIGHT_GI
- constexpr std::string_view serif::utils::misc::terminal::a_reset = resolve_ansi_color(ANSIColors::RESET)

11.230 src/include/serif/utils/misc/concepts/type_uniqueness.hpp File Reference

#include "serif/utils/misc/traits/type_uniqueness.hpp"
Include dependency graph for type_uniqueness.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

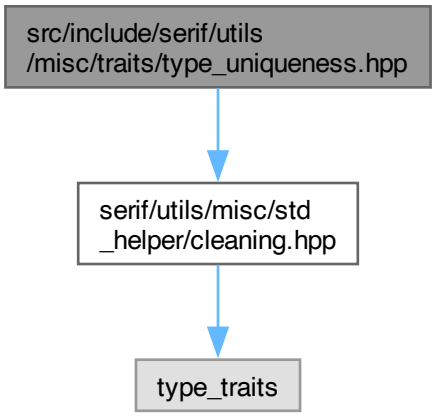
- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::concepts

Concepts

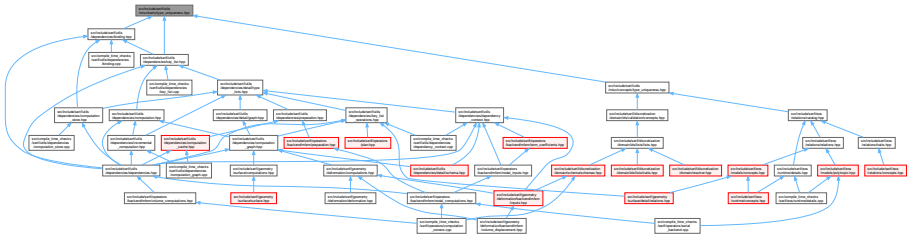
- concept serif::utils::misc::concepts::AllTypesAreUnique

11.231 src/include/serif/utils/misc/traits/type_uniqueness.hpp File Reference

#include "serif/utils/misc/std_helper/cleaning.hpp"
Include dependency graph for type_uniqueness.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

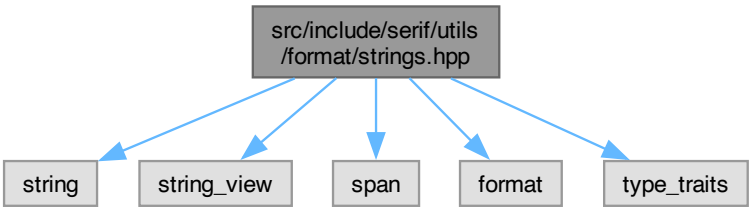
- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::traits

Variables

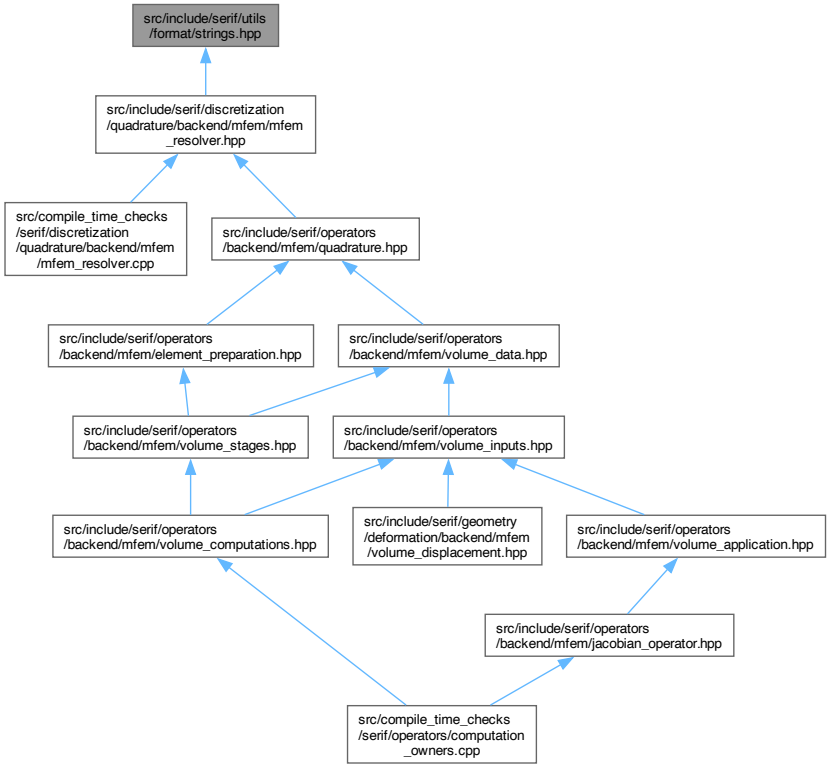
- template<typename... Ts>
  constexpr bool serif::utils::misc::traits::all_unique_types_v = true
- template<typename First, typename... Rest>
  constexpr bool serif::utils::misc::traits::all_unique_types_v< First, Rest... > = (!std_helper::SameType<First, Rest> && ...) && all_unique_types_v<Rest...>

11.232 src/include/serif/utils/format/strings.hpp File Reference

#include <string>
#include <string_view>
#include
#include <format>
#include <type_traits>
Include dependency graph for strings.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::misc
- namespace serif::utils::misc::format

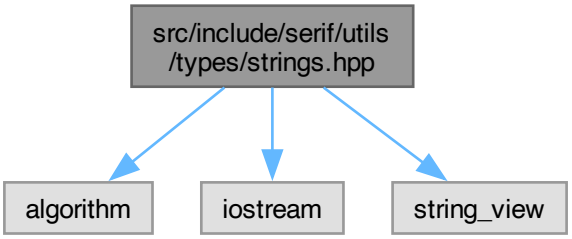
Functions

- template<typename T>
  std::string serif::utils::misc::format::join (const std::span< T > &elements, std::string_view delimiter)

11.233 src/include/serif/utils/types/strings.hpp File Reference

#include <algorithm>
#include <iostream>

#include <string_view>
Include dependency graph for strings.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct serif::utils::types::ctstring< N >
- struct serif::utils::types::ct_error_string< N >

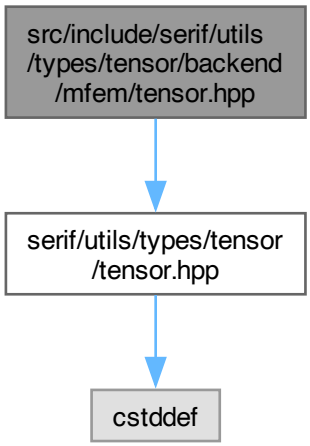
Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::types

11.234 src/include/serif/utils/types/tensor/backend/mfem/tensor.hpp File Reference

#include "serif/utils/types/tensor/tensor.hpp"

Include dependency graph for tensor.hpp:

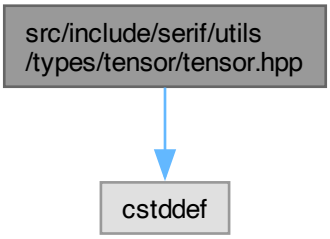


Namespaces

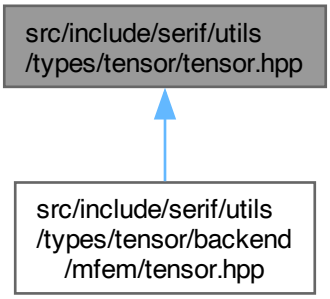
- namespace serif
- namespace serif::utils
- namespace serif::utils::types
- namespace serif::utils::types::backend
- namespace serif::utils::types::backend::mfem

11.235 src/include/serif/utils/types/tensor/tensor.hpp File Reference

#include <cstddef>
Include dependency graph for tensor.hpp:



This graph shows which files directly or indirectly include this file:



Classes

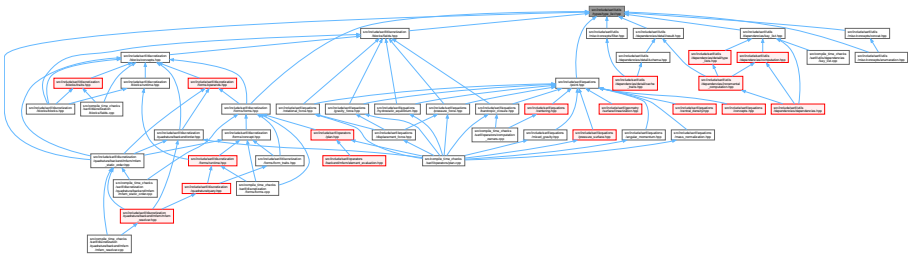
- struct serif::utils::types::VectorBase< T >
- struct serif::utils::types::DenseMatrixBase< T >
- struct serif::utils::types::SparseMatrixBase< T >
- struct serif::utils::types::DenseTensorBase< T, Rank >
- struct serif::utils::types::SparseTensorBase< T, Rank >

Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::types

11.236 src/include/serif/utils/types/type_list.hpp File Reference

This graph shows which files directly or indirectly include this file:



Classes

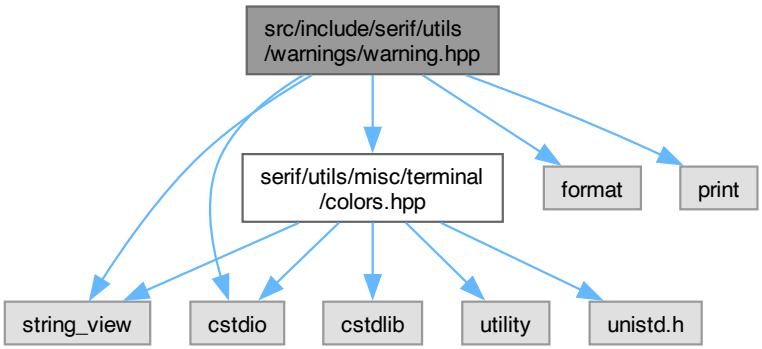
- struct serif::utils::types::TypeList< Ts >

Namespaces

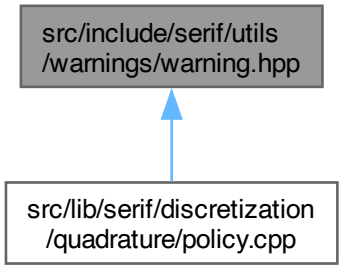
- namespace serif
- namespace serif::utils
- namespace serif::utils::types

11.237 src/include/serif/utils/warnings/warning.hpp File Reference

#include <string_view>
#include <format>
#include <print>
#include <cstdio>
#include "serif/utils/misc/terminal/colors.hpp"
Include dependency graph for warning.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace serif
- namespace serif::utils
- namespace serif::utils::warnings

Macros

- #define DISABLE_WARNING 0

Functions

- void serif::utils::warnings::serif_warning (const std::string_view message, auto... args)

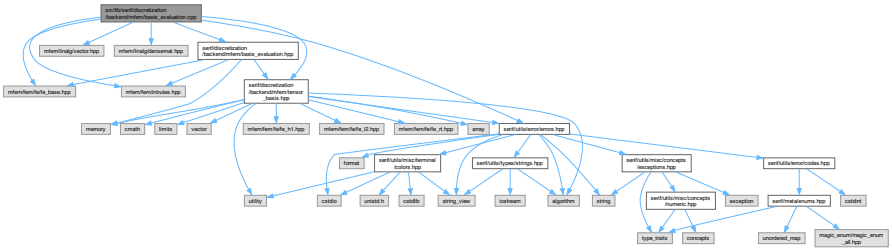
11.237.1 Macro Definition Documentation

11.237.1.1 DISABLE_WARNING

#define DISABLE_WARNING 0

11.238 src/lib/serif/discretization/backend/mfem/basis_evaluation.cpp File Reference

#include "mfem/fem/fe_base.hpp"
#include "mfem/fem/intrules.hpp"
#include "mfem/linalg/vector.hpp"
#include "mfem/linalg/densemat.hpp"
#include "serif/discretization/backend/mfem/basis_evaluation.hpp"
#include "serif/discretization/backend/mfem/tensor_basis.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for basis_evaluation.cpp:



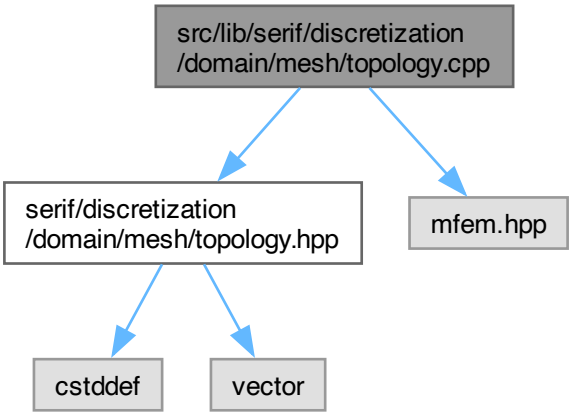
Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::backend
- namespace serif::discretization::backend::mfem

11.239 src/lib/serif/discretization/domain/mesh/topology.cpp File Reference

#include "serif/discretization/domain/mesh/topology.hpp"
#include <mfem.hpp>

Include dependency graph for topology.cpp:



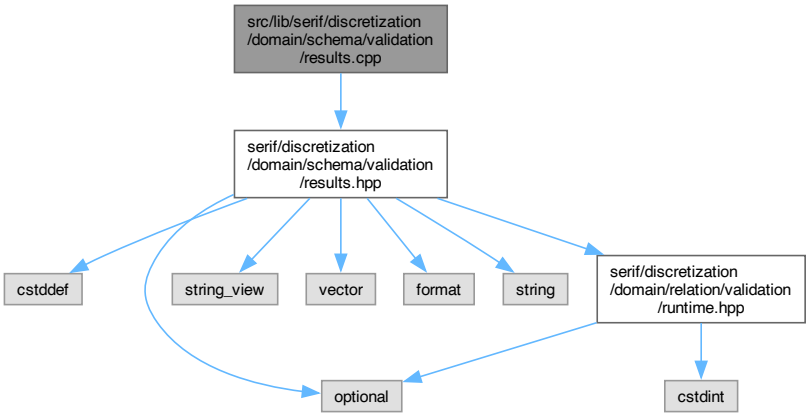
Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::mesh

11.240 src/lib/serif/discretization/domain/schema/validation/results.cpp File Reference

#include ”serif/discretization/domain/schema/validation/results.hpp”

Include dependency graph for results.cpp:

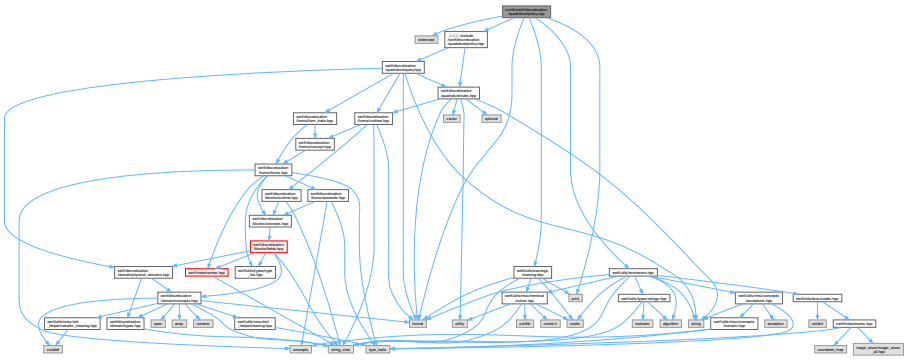


Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::domain
- namespace serif::discretization::domain::schema
- namespace serif::discretization::domain::schema::validation

11.241 src/lib/serif/discretization/quadrature/policy.cpp File Reference

#include <stdexcept>
#include <format>
#include <print>
#include "../../../../include/serif/discretization/quadrature/policy.hpp"
#include "serif/utils/warnings/warning.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for policy.cpp:

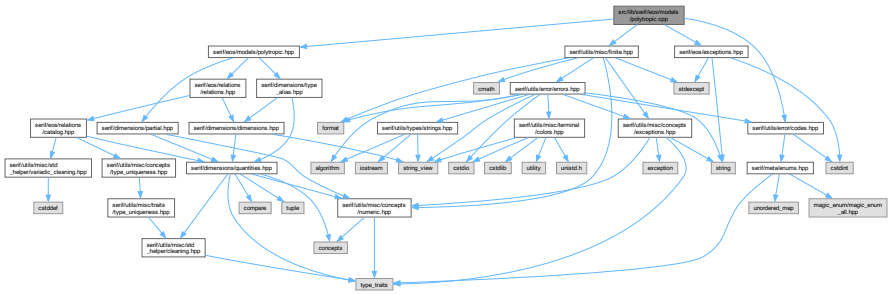


Namespaces

- namespace serif
- namespace serif::discretization
- namespace serif::discretization::quadrature

11.242 src/lib/serif/eos/models/polytropic.cpp File Reference

#include ”serif/eos/models/polytropic.hpp”
#include ”serif/utils/misc/finite.hpp”
#include ”serif/eos/exceptions.hpp”
#include ”serif/utils/error/codes.hpp”
Include dependency graph for polytropic.cpp:



Namespaces

- namespace serif
- namespace serif::eos
- namespace serif::eos::models

Enumerations

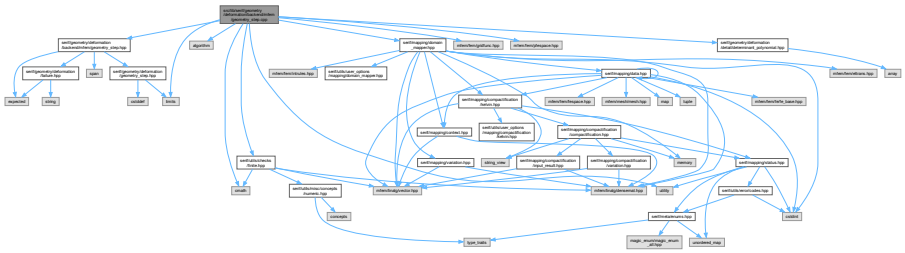
- enum class serif::eos::models::SERiFErrorCode

Functions

- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsNumeric Scalar>
  void serif::eos::models::validate_finite (const Scalar value)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void serif::eos::models::validate_greater_than (const Scalar value, const Scalar predicate)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void serif::eos::models::validate_less_than (const Scalar value, const Scalar predicate)
- template<typename ExceptionType, errors::SERiFErrorCode ErrorCode, concepts::IsComparableToNumeric Scalar>
  void serif::eos::models::validate_nonnegativity (const Scalar value)

11.243 src/lib/serif/geometry/deformation/backend/mfem/geometry_↩ step.cpp File Reference

#include ”serif/geometry/deformation/backend/mfem/geometry_step.hpp”
#include <algorithm>
#include <cmath>
#include <limits>
#include <utility>
#include ”mfem/fem/gridfunc.hpp”
#include ”mfem/fem/pfespace.hpp”
#include ”serif/geometry/deformation/detail/determinant_polynomial.hpp”
#include ”serif/mapping/domain_mapper.hpp”
#include ”serif/utils/checks/finite.hpp”
Include dependency graph for geometry_step.cpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

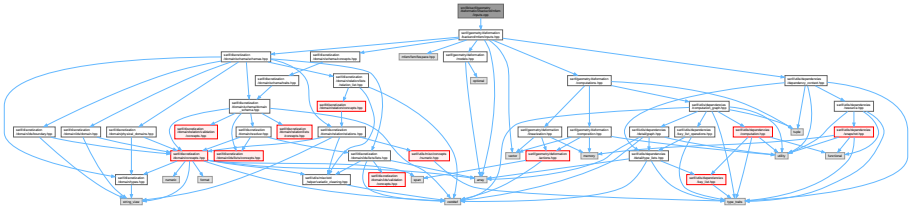
Functions

- std::expected< GeometryStepLimit, DeformationFailure > serif::geometry::deformation::backend::mfem::estimate_geon (const mapping::DomainMapper &mapper, const ::mfem::FiniteElementSpace &displacement_↩ space, const ::mfem::GridFunction &compactification_coordinate, const ::mfem::Vector &accepted_↩ displacement, const ::mfem::Vector &displacement_direction, std::span< const GeometrySampleRule > sample_rules, GeometryStepOptions options={})

  Find the first sampled mapping boundary along accepted + step direction.

11.244 src/lib/serif/geometry/deformation/backend/mfem/inputs.cpp File Reference

#include "serif/geometry/deformation/backend/mfem/inputs.hpp"
Include dependency graph for inputs.cpp:

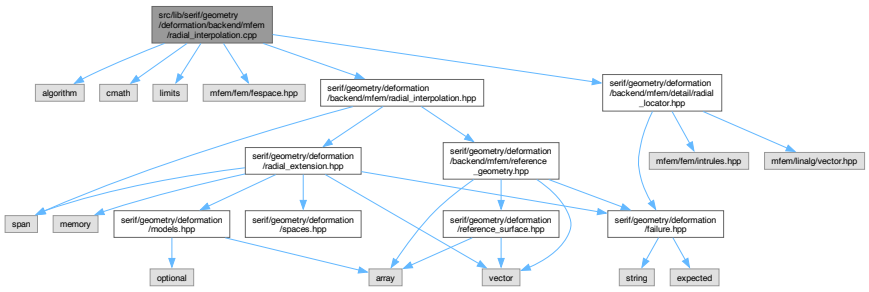


Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

11.245 src/lib/serif/geometry/deformation/backend/mfem/radial_↩ interpolation.cpp File Reference

#include <algorithm>
#include <cmath>
#include <limits>
#include "mfem/fem/fespace.hpp"
#include "serif/geometry/deformation/backend/mfem/radial_interpolation.hpp"
#include "serif/geometry/deformation/backend/mfem/detail/radial_locator.hpp"
Include dependency graph for radial_interpolation.cpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

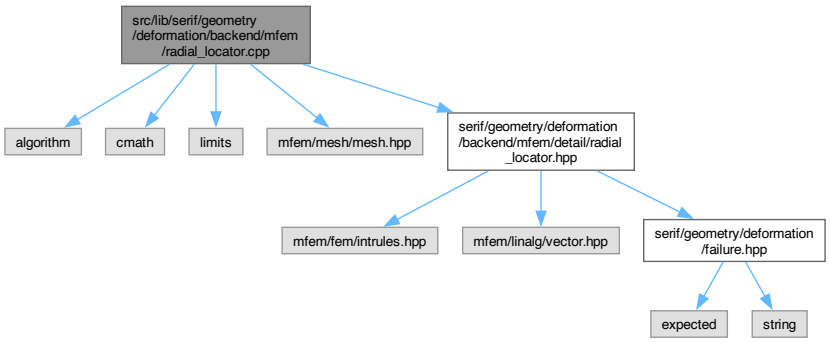
Functions

- std::expected< RadialInterpolation, DeformationFailure > serif::geometry::deformation::backend::mfem::build_radial_i (const ::mfem::FiniteElementSpace &scalar_space, const ::mfem::FiniteElementSpace &vector_space, ::mfem::Mesh &logical_mesh, std::span< const int > stellar_element_support, std::span< const int > vacuum_element_support, int surface_attribute, int infinity_attribute)

  Compile logical-ray interpolation on paired physical and logical meshes. Supports are zero/one element masks. The logical radius is the maximum absolute Cartesian coordinate, preserving the STROID reference foliation after physical distortion. Only conforming serial nodal spaces and component-block vector ordering are supported.

11.246 src/lib/serif/geometry/deformation/backend/mfem/radial_↩ locator.cpp File Reference

#include <algorithm>
#include <cmath>
#include <limits>
#include "mfem/mesh/mesh.hpp"
#include "serif/geometry/deformation/backend/mfem/detail/radial_locator.hpp"
Include dependency graph for radial_locator.cpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem
- namespace serif::geometry::deformation::backend::mfem::detail

Functions

- double serif::geometry::deformation::backend::mfem::detail::logical_radius (const ::mfem::Vector &position)

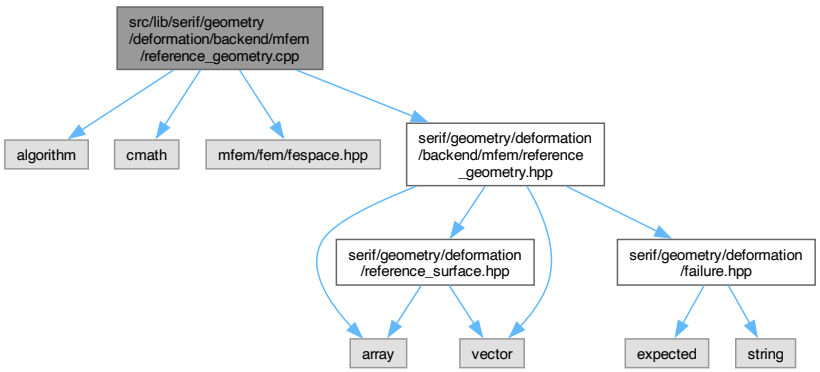
  Preserve the cubical logical radius used by the original radial foliation.
- std::expected< double, DeformationFailure > serif::geometry::deformation::backend::mfem::detail::boundary_logical_ra (::mfem::Mesh &mesh, int attribute)

  Check a complete cubical shell and return its logical radius.
- std::expected< LocatedBoundaryPoint, DeformationFailure > serif::geometry::deformation::backend::mfem::detail::locat (::mfem::Mesh &mesh, int attribute, const ::mfem::Vector &target)

  Locate a projected logical ray on a quadrilateral shell, preserving MeanField tolerances.

11.247 src/lib/serif/geometry/deformation/backend/mfem/reference_↩ geometry.cpp File Reference

#include <algorithm>
#include <cmath>
#include "mfem/fem/fespace.hpp"
#include "serif/geometry/deformation/backend/mfem/reference_geometry.hpp"
Include dependency graph for reference_geometry.cpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

Functions

- DeformationStatus serif::geometry::deformation::backend::mfem::validate_scalar_reference_space (const ::mfem::FiniteElementSpace &scalar_space)

  Reject spaces whose coefficients cannot be interpreted as serial nodal positions.
- DeformationStatus serif::geometry::deformation::backend::mfem::validate_companion_numbering (const ::mfem::FiniteElementSpace &reference_space, const ::mfem::FiniteElementSpace &companion↩ _space)

  Require identical scalar coefficient meanings in two companion spaces. Vector component ordering may differ; element bases and their scalar-node indices must agree.
- std::expected< std::vector< std::array< double, 3 > >, DeformationFailure > serif::geometry::deformation::backend::mfe (const ::mfem::FiniteElementSpace &scalar_space, ::mfem::Mesh &reference_mesh)

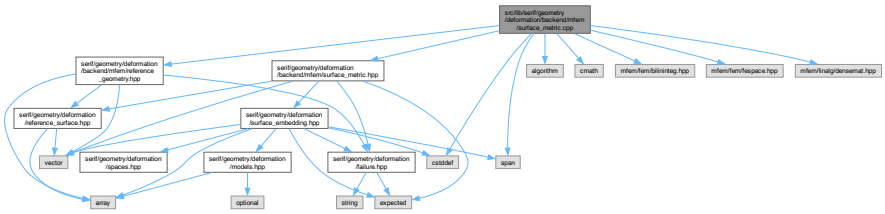
  Sample a paired mesh at the scalar companion nodes, checking shared-node agreement.
- std::expected< std::vector< int >, DeformationFailure > serif::geometry::deformation::backend::mfem::select_surface_s (const ::mfem::FiniteElementSpace &scalar_space, int surface_attribute)

  Select canonical surface nodes without retaining a mesh or finite-element-space pointer.
- std::expected< ReferenceSurface, DeformationFailure > serif::geometry::deformation::backend::mfem::sample_reference (const ::mfem::FiniteElementSpace &scalar_space, int surface_attribute)

  Sample physical reference positions in ascending scalar-DOF order.

11.248 src/lib/serif/geometry/deformation/backend/mfem/surface_↩ metric.cpp File Reference

#include "serif/geometry/deformation/backend/mfem/surface_metric.hpp"
#include <algorithm>
#include <cmath>
#include <cstddef>
#include
#include "mfem/fem/bilininteg.hpp"
#include "mfem/fem/fespace.hpp"
#include "mfem/linalg/densemat.hpp"
#include "serif/geometry/deformation/backend/mfem/reference_geometry.hpp"
Include dependency graph for surface_metric.cpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::backend
- namespace serif::geometry::deformation::backend::mfem

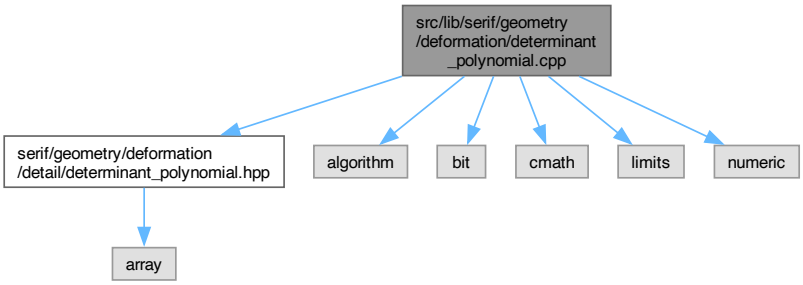
Functions

- std::expected< std::vector< double >, DeformationFailure > serif::geometry::deformation::backend::mfem::surface_para (const ::mfem::FiniteElementSpace &scalar_space, int surface_attribute, const ReferenceSurface &reference_surface, const SurfaceEmbedding &embedding)

  Measure each shape parameter with the reference boundary mass norm of its Cartesian finite-element trace.

11.249 src/lib/serif/geometry/deformation/determinant_polynomial.cpp File Reference

#include "serif/geometry/deformation/detail/determinant_polynomial.hpp"
#include <algorithm>
#include <bit>
#include <cmath>
#include <limits>
#include <numeric>
Include dependency graph for determinant_polynomial.cpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::detail

Functions

- DeterminantPolynomial serif::geometry::deformation::detail::determinant_polynomial (const JacobianEntries &base, const JacobianEntries &direction, int dimension, double maximum_step, double determinant↩ _floor) noexcept

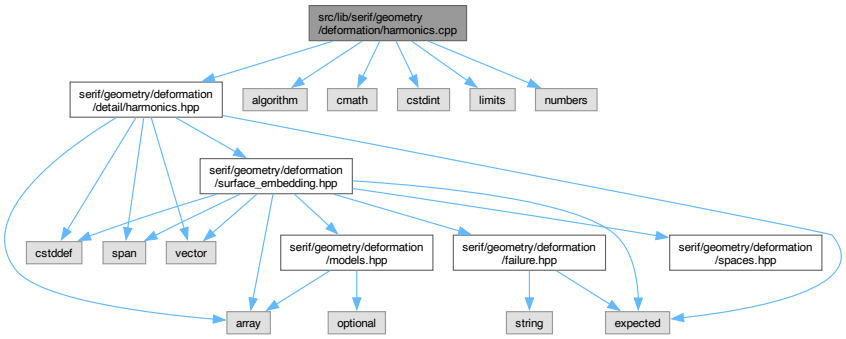
  Expand det(base + t maximum_step direction) minus the floor for dimensions one through three.
- double serif::geometry::deformation::detail::evaluate_polynomial (const DeterminantPolynomial &coefficients, double parameter) noexcept

  Evaluate the cubic with fused operations, retaining small values near repeated roots.
- double serif::geometry::deformation::detail::first_boundary_parameter (const DeterminantPolynomial &coefficients) noexcept

  Find the first loss of strict positivity in (0, 1], including a tangent zero.

11.250 src/lib/serif/geometry/deformation/harmonics.cpp File Reference

#include ”serif/geometry/deformation/detail/harmonics.hpp”
#include <algorithm>
#include <cmath>
#include <cstdint>
#include <limits>
#include <numbers>
Include dependency graph for harmonics.cpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation
- namespace serif::geometry::deformation::detail

Functions

- std::expected< std::vector< SphericalHarmonicMode >, DeformationFailure > serif::geometry::deformation::detail::mak (const SphericalHarmonicSurface &model, std::size_t node_count)

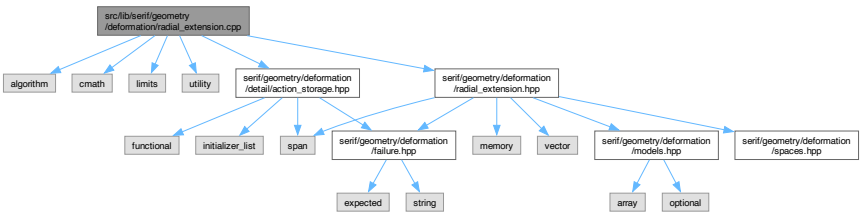
  Validate harmonic cutoffs before allocating a basis or enumerating modes.
- std::expected< std::vector< double >, DeformationFailure > serif::geometry::deformation::detail::sample_harmonic_ba (const SphericalHarmonicSurface &model, std::span< const SphericalHarmonicMode > modes, std↩ ::span< const std::array< double, 3 > > directions)

  Use normalized Legendre recurrence to avoid factorial growth while preserving the Condon–Shortley phase.
- DeformationStatus serif::geometry::deformation::detail::check_independent_harmonics (std::span< const double > basis, std::size_t node_count, std::size_t mode_count)

  Check sampled rank on a private copy, leaving physical coefficients and basis normalization unchanged.

11.251 src/lib/serif/geometry/deformation/radial_extension.cpp File Reference

#include <algorithm>
#include <cmath>
#include <limits>
#include <utility>
#include "serif/geometry/deformation/radial_extension.hpp"
#include "serif/geometry/deformation/detail/action_storage.hpp"
Include dependency graph for radial_extension.cpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

Functions

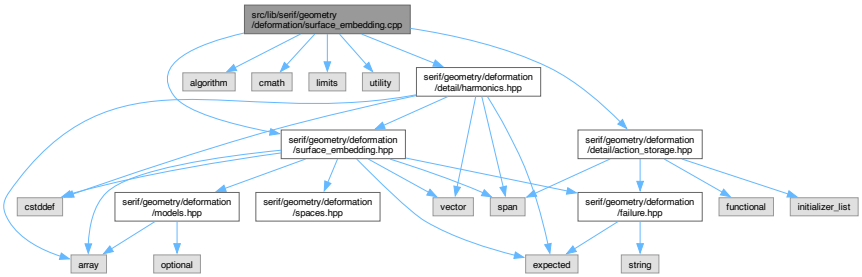
- std::expected< RadialExtension, DeformationFailure > serif::geometry::deformation::make_radial_extension (std::shared_ptr< const RadialInterpolation > interpolation, PowerLawRadialExtension interior={}, FixedInfinityRadialExtension vacuum={})

  Compile regional weights while sharing the immutable surface interpolation. Callers supplying a shared pointer must treat its data as immutable after compilation; the dependency graph publishes a const owning value for precisely this reason.

11.252 src/lib/serif/geometry/deformation/surface_embedding.cpp File Reference

#include "serif/geometry/deformation/surface_embedding.hpp"
#include <algorithm>
#include <cmath>
#include <limits>
#include <utility>
#include "serif/geometry/deformation/detail/harmonics.hpp"

#include ”serif/geometry/deformation/detail/action_storage.hpp”
Include dependency graph for surface_embedding.cpp:



Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::deformation

Functions

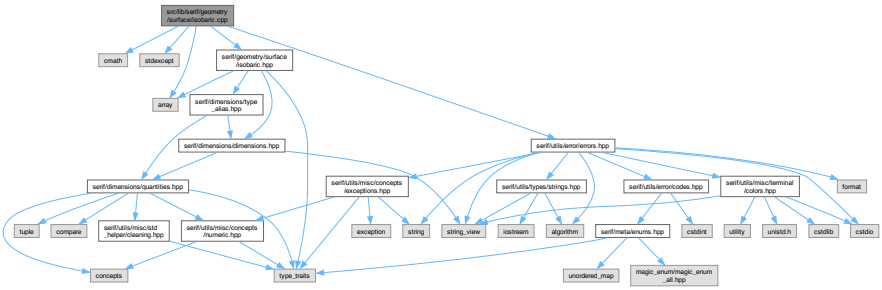
- std::expected< SurfaceEmbedding, DeformationFailure > serif::geometry::deformation::build_surface_embedding (const NodalRadialSurface &model, std::span< const std::array< double, 3 > > positions)

  Compile fixed radial directions from ordered physical reference positions without retaining backend objects.
- std::expected< SurfaceEmbedding, DeformationFailure > serif::geometry::deformation::build_surface_embedding (const SphericalHarmonicSurface &model, std::span< const std::array< double, 3 > > positions)

  Sample physically normalized real harmonics and reject unresolved modes without renormalizing or dropping coordinates.

11.253 src/lib/serif/geometry/surface/isobaric.cpp File Reference

#include <cmath>
#include <stdexcept>
#include <array>
#include ”serif/geometry/surface/isobaric.hpp”
#include ”serif/utils/error/errors.hpp”
Include dependency graph for isobaric.cpp:

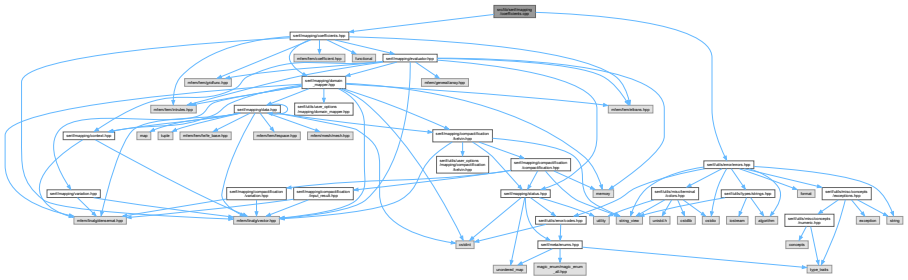


Namespaces

- namespace serif
- namespace serif::geometry
- namespace serif::geometry::surface

11.254 src/lib/serif/mapping/coefficients.cpp File Reference

#include ”serif/mapping/coefficients.hpp”
#include ”serif/utils/error/errors.hpp”
Include dependency graph for coefficients.cpp:

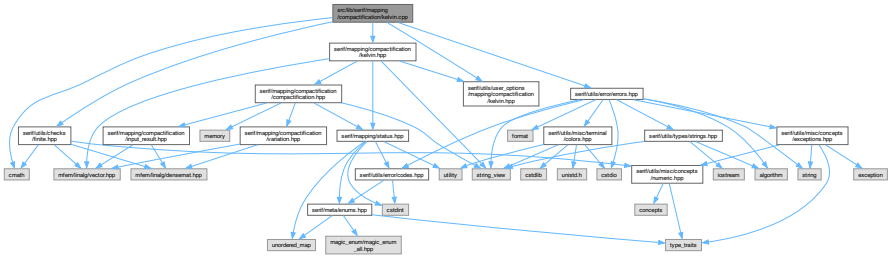


Namespaces

- namespace serif
- namespace serif::mapping

11.255 src/lib/serif/mapping/compactification/kelvin.cpp File Reference

#include <cmath>
#include ”serif/utils/checks/finite.hpp”
#include ”serif/mapping/compactification/kelvin.hpp”
#include ”serif/utils/user_options/mapping/compactification/kelvin.hpp”
#include ”serif/utils/error/errors.hpp”
Include dependency graph for kelvin.cpp:

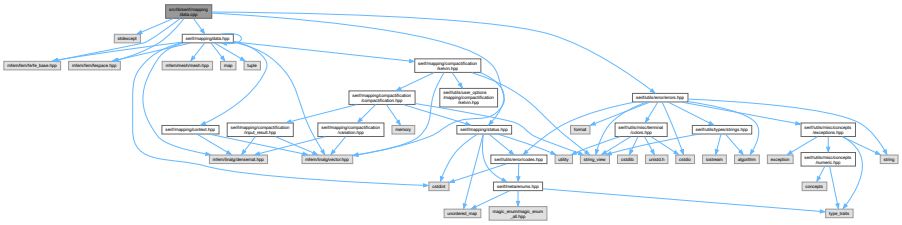


Namespaces

- namespace serif
- namespace serif::mapping
- namespace serif::mapping::compactification

11.256 src/lib/serif/mapping/data.cpp File Reference

#include <stdexcept>
#include "mfem/linalg/vector.hpp"
#include "mfem/fem/fe/fe_base.hpp"
#include "mfem/fem/fespace.hpp"
#include "serif/mapping/data.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for data.cpp:



Namespaces

- namespace serif
- namespace serif::mapping

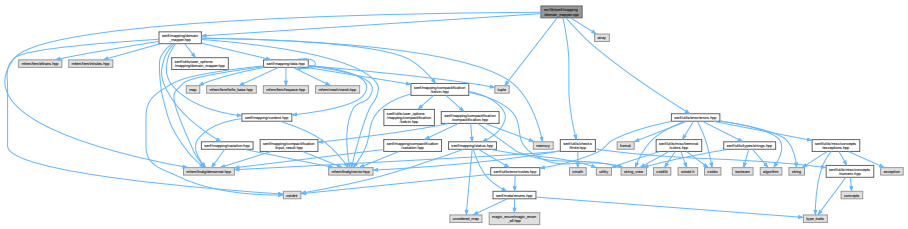
Functions

- ElementDisplacementData serif::mapping::element_displacement_data_from_element_vdofs (const mfem::FiniteElement &element, mfem::Vector &displacement_dofs)

11.257 src/lib/serif/mapping/domain_mapper.cpp File Reference

#include "mfem/linalg/densemat.hpp"
#include "serif/mapping/domain_mapper.hpp"
#include "serif/utils/error/errors.hpp"
#include "serif/utils/checks/finite.hpp"
#include <array>

#include <tuple>
Include dependency graph for domain_mapper.cpp:

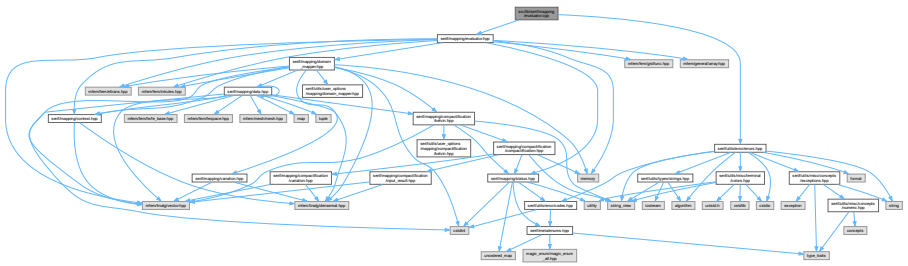


Namespaces

- namespace serif
- namespace serif::mapping

11.258 src/lib/serif/mapping/evaluator.cpp File Reference

#include "serif/mapping/evaluator.hpp"
#include "serif/utils/error/errors.hpp"
Include dependency graph for evaluator.cpp:



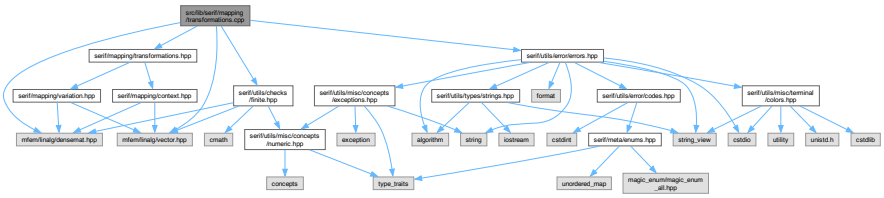
Namespaces

- namespace serif
- namespace serif::mapping

11.259 src/lib/serif/mapping/transformations.cpp File Reference

#include "serif/mapping/transformations.hpp"
#include "serif/utils/error/errors.hpp"
#include "mfem/linalg/vector.hpp"
#include "mfem/linalg/densemat.hpp"

#include "serif/utils/checks/finite.hpp"
Include dependency graph for transformations.cpp:



Namespaces

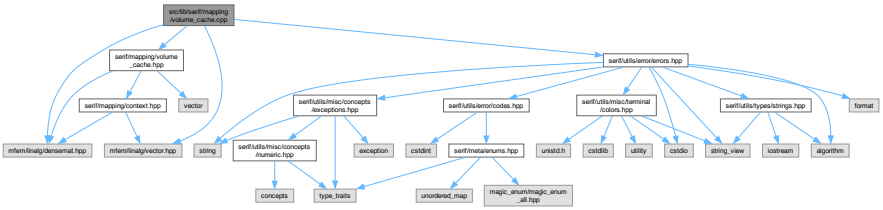
- namespace serif
- namespace serif::mapping

Functions

- void serif::mapping::map_hdiv_flux_to_physical (const MappingPointContext &context, const mfem::Vector &reference_flux, mfem::Vector &physical_flux)
- void serif::mapping::map_physical_flux_to_hdiv_reference (const MappingPointContext &context, const mfem::Vector &physical_flux, mfem::Vector &reference_flux)
- void serif::mapping::map_reference_gradient_to_physical (const MappingPointContext &context, const mfem::Vector &reference_gradient, mfem::Vector &physical_gradient)
- void serif::mapping::map_physical_gradient_to_reference (const MappingPointContext &context, const mfem::Vector &physical_gradient, mfem::Vector &reference_gradient)
- void serif::mapping::map_reference_vector_gradient_to_physical (const MappingPointContext &context, const mfem::DenseMatrix &reference_vector_gradient, mfem::DenseMatrix &physical_vector_↩ gradient)
- void serif::mapping::map_physical_vector_gradient_to_reference (const MappingPointContext &context, const mfem::DenseMatrix &physical_vector_gradient, mfem::DenseMatrix &reference_vector_↩ gradient)
- double serif::mapping::map_hdiv_divergence_to_physical (const MappingPointContext &context, double reference_divergence)
- void serif::mapping::compute_hdiv_mass_tensor (const MappingPointContext &context, mfem::↩ DenseMatrix &mass_tensor)
- void serif::mapping::compute_scalar_diffusion_tensor (const MappingPointContext &context, mfem↩ ::DenseMatrix &diffusion_tensor)
- void serif::mapping::map_hcurl_field_to_physical (const MappingPointContext &context, const mfem::Vector &reference_field, mfem::Vector &physical_field)
- void serif::mapping::map_physical_field_to_hcurl_reference (const MappingPointContext &context, const mfem::Vector &physical_field, mfem::Vector &reference_field)
- void serif::mapping::map_hcurl_curl_to_physical (const MappingPointContext &context, const mfem::Vector &reference_curl, mfem::Vector &physical_curl)
- void serif::mapping::map_physical_curl_to_hcurl_reference (const MappingPointContext &context, const mfem::Vector &physical_curl, mfem::Vector &reference_curl)
- void serif::mapping::compute_hdiv_mass_tensor_variation (const MappingPointContext &context, const MappingPointVariation &variation, mfem::DenseMatrix &mass_tensor_variation)

11.260 src/lib/serif/mapping/volume_cache.cpp File Reference

#include ”serif/mapping/volume_cache.hpp”
#include ”serif/utils/error/errors.hpp”
#include ”mfem/linalg/densemat.hpp”
#include ”mfem/linalg/vector.hpp”
Include dependency graph for volume_cache.cpp:



Namespaces

- namespace serif
- namespace serif::mapping

Index

@1200271712111733323463461531470020022660652232665::HasBoundType, 221
@1200271712111733323463461531470020022660652232665::HasBoundTypeMember, 221
@2000743173031670560630642560700462030540640522145::CanInspectChange, 221
@2000743173031670560630642560700462030540640522145::CanMakeCache, 222
@2000743173031670560630642560700462030540640522145::CanMakeComputation, 222
@2000743173031670560630642560700462030540640522145::CanMakeFallibleCache, 222
@2000743173031670560630642560700462030540640522145::CanRequest, 222
@2303720751251171121462160220502573311771371073277::CanCompose, 222
@2470062770772112161533503411423573371610010261437::HasSurfaceRelation, 223
@2470062770772112161533503411423573371610010261437::SelectsCarrier, 223
@2521661430352643140212142460260550411201633220417::CanResolve, 223
@3231723273641022352103333370570650151523543313437::CanBorrow, 223
@3231723273641022352103333370570650151523543313437::CanMakeContext, 224
@3231723273641022352103333370570650151523543313437::CanRead, 224
@3231723273641022352103333370570650151523543313437::CanSelect, 224
@3471452410341021410720043571712743602303460643567::CanMakeStore, 224
@3471452410341021410720043571712743602303460643567::CanOwn, 224
@3471452410341021410720043571712743602303460643567::CanRead, 225
@3471452410341021410720043571712743602303460643567::CanReadEntry, 225
~DofMap
serif::discretization::backend::mfem::DofMap, 440
~DomainMapper
serif::mapping::DomainMapper, 469
~ExteriorDomainMap
serif::mapping::compactification::ExteriorDomainMap, 525
a_black
serif::utils::misc::terminal, 217
a_blue
serif::utils::misc::terminal, 217
a_bright_black
serif::utils::misc::terminal, 217
a_bright_green
serif::utils::misc::terminal, 217
a_bright_red
serif::utils::misc::terminal, 218
a_cyan
serif::utils::misc::terminal, 218
a_green
serif::utils::misc::terminal, 218
a_magenta
serif::utils::misc::terminal, 218
a_red
serif::utils::misc::terminal, 218
a_white
serif::utils::misc::terminal, 218
a_yellow
serif::utils::misc::terminal, 218
AccelerationValue
serif::dimensions, 62
accumulate_element_result
serif::operators::backend::mfem, 173
accumulate_partial
serif::geometry::surface::detail, 144
actualBoundaryID
serif::discretization::domain::relation::validation::RelationValidati 451
serif::discretization::domain::schema::validation::RelationValidati 455
acyclic
serif::utils::dependencies::detail::Graph< Computations >, 606
add
serif::utils::CompensatedSum, 357
add_element
serif::discretization::backend::mfem::DofMap, 440

add_jacobian_transpose_rows
serif::operators::backend::mfem, 173
add_point_border_residual
serif::operators::backend::mfem, 173
additional_boost
serif::discretization::quadrature::QuadratureTermOptions, 810
additional_order
serif::equations::AngularMomentum, 267
serif::equations::BarotropicClosure< EOST >, 284
serif::equations::BarotropicHydrostaticEquilibrium, 287
serif::equations::gravity_contributions::DensitySource, 412
serif::equations::gravity_contributions::GradientDivergence, 600
serif::equations::gravity_contributions::GradientValue, 604
serif::equations::gravity_contributions::PotentialValue, 775
serif::equations::GravityForce, 613
serif::equations::MassNormalization, 694
serif::equations::PressureForce< EOST >, 789
serif::equations::RotationalForce, 890
serif::operators::BarotropicHydrostaticEquilibrium, 289
serif::operators::GradientRequest, 602
adjacentDomainID
serif::discretization::domain::relation::validation::RelationValidation, 637
serif::discretization::domain::schema::validation::RelationValidation, 638
adjacentElementID
serif::discretization::domain::relation::validation::RelationValidation, 637
serif::discretization::domain::schema::validation::RelationValidation, 638
affine
serif::discretization::quadrature, 103
serif::equations::CentralDensityConstraint< EOST >, 342
agrees_with
serif::discretization::backend::mfem::TensorBasis, 951
AIsBaseClassOfB
serif::discretization::domain, 79
serif::discretization::domain::relation, 85
serif::utils::misc::std_helper, 215
all_unique_types_v
serif::utils::misc::traits, 219
all_unique_types_v< First, Rest... >
serif::utils::misc::traits, 219
all_unique_values
serif::discretization::domain::ids::validation, 84
AllDomains
serif::discretization::domain, 78
AllInputs
serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > >, 845
angular_velocity
serif::equations::RigidRotation, 883
angular_velocity_term
serif::discretization::blocks::rotation, 887
AngularMomentum
serif::equations::AngularMomentum, 267
AngularMomentumValue
serif::dimensions, 62
AngularVelocity
serif::equations::AngularMomentum, 266
serif::equations::BarotropicHydrostaticEquilibrium, 286
serif::equations::RotationalForce, 889
serif::operators::BarotropicHydrostaticEquilibrium, 288
AngularVelocityValue
serif::dimensions, 62
serif::equations::AngularMomentum, 266
serif::equations::RotationalForce, 889
ANSIColors
serif::utils::misc::terminal, 216
apply
serif::discretization::backend::mfem::TensorBasis, 952
serif::geometry::deformation::InscribedDiagnostics, 824
serif::geometry::deformation::SurfaceEmbedding, 841
serif::geometry::deformation::VolumeEmbedding, 922
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 360
serif::operators::backend::mfem, 173, 174
apply_jacobian_rows
serif::operators::backend::mfem, 174
apply_jacobian_transpose
serif::geometry::deformation, 123
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 360
serif::operators::backend::mfem, 174
apply_point_border_jacobian
serif::operators::backend::mfem, 175
apply_point_border_jacobian_transpose
serif::operators::backend::mfem, 175
apply_pullback_derivative
serif::geometry::deformation, 123

serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 360
apply_residual_rows
serif::operators::backend::mfem, 175
apply_transpose
serif::geometry::deformation::RadialExtension, 824
serif::geometry::deformation::SurfaceEmbedding, 941
apply_volume_direction
serif::operators::backend::mfem, 175
AreaValue
serif::dimensions, 62
AT_COMPACTIFIED_INFINITY
serif::mapping, 150
serif::utils::errors, 209
automatic
serif::discretization::backend::mfem, 70

BarotropicClosure
serif::equations::BarotropicClosure< EOST >, 284
BarotropicHydrostaticEquilibrium
serif::equations::BarotropicHydrostaticEquilibrium, 286
serif::operators::BarotropicHydrostaticEquilibrium, 289
base_order
serif::discretization::quadrature::Query, 820
serif::discretization::quadrature::Resolution, 870
basis
serif::discretization::backend::mfem::TensorBasisCache::Entry, 500
serif::operators::backend::mfem::ElementData< OperandCount >, 487
serif::operators::backend::mfem::TensorBasisCache::Entry, 501
serif::operators::backend::mfem::VolumeScratch, 1023
serif::operators::backend::mfem::Workspace, 1027
begin
serif::discretization::quadrature::RuleSet, 892
bind
serif::operators::backend::mfem, 176
bind_contributions
serif::operators, 159
serif::operators::detail, 193
binding
serif::discretization::backend::mfem::DofSelection, 447
serif::operators::backend::mfem::CoefficientView< BindingT >, 354
serif::operators::backend::mfem::MappingCoefficients, 676
Bindings
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 367
serif::utils::dependencies::ComputationStore< EntryTs >, 381
serif::utils::dependencies::DependencyContext< ResourceTs >, 413
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >, 416
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >, 375
bindings
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >, 1011
BLACK
serif::utils::misc::terminal, 217
BLUE
serif::utils::misc::terminal, 217
boost
serif::discretization::quadrature::Resolution, 870
serif::discretization::quadrature::RuleControl, 891
Border
serif::equations::CentralDensityConstraint< EOST >, 341
serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >, 760
border_binding
serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >, 760
serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >, 761
border_coefficient
serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >, 760
border_term
serif::discretization::blocks::central_density, 339
BorderBinding
serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >, 761
borrow
serif::utils::dependencies, 199
bound_type_t
serif::utils::dependencies, 198
boundary_attributes
serif::discretization::backend::mfem::DofSelection, 447
serif::geometry::deformation::backend::mfem::ReferenceMeshSnap 836
boundary_descriptors
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 479

boundary_element
serif::geometry::deformation::backend::mfem::detail, 673
boundary_element_boundary_id
serif::discretization::domain::mesh::MeshTopology, 699
boundary_element_count
serif::discretization::domain::mesh::MeshTopology, 699
boundary_element_face
serif::discretization::domain::mesh::MeshTopology, 699
boundary_elements_on_face
serif::discretization::domain::mesh::MeshTopology, 700
boundary_id
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 479
boundary_id_count
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 480
boundary_id_matches
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 479
boundary_ids_type
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 478
boundary_logical_radius
serif::geometry::deformation::backend::mfem::detail, 133
boundary_step
serif::geometry::deformation::GeometryStepLimit, 593
boundary_type
serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >, 320
serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >, 451
serif::discretization::domain::schema::BoundaryID< BoundaryT, IdentifierValue >, 321
serif::discretization::domain::schema::validation::DomainBoundary< BoundaryT, DomainTs >, 453
boundary_values
serif::geometry::surface, 139
boundaryElementID
serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics, 454
serif::discretization::domain::schema::validation::RelationValidationResult::DomainBoundaryDiagnostics, 455
BoundTerms
serif::geometry::surface::detail, 143
BRIGHT_BLACK
serif::utils::misc::terminal, 217
BRIGHT_GREEN
serif::utils::misc::terminal, 217
BRIGHT_RED
serif::utils::misc::terminal, 217
build_displacement
serif::geometry::deformation, 124
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 361
build_radial_interpolation
serif::geometry::deformation::backend::mfem, 128
build_surface_embedding
serif::geometry::deformation, 124
serif::geometry::deformation::SurfaceEmbedding, 943

CACHE_FE_BINDING_ERROR
serif::utils::errors, 210
capture
serif::discretization::backend::mfem::CoefficientTracking, 348, 349
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza 834
serif::geometry::surface::detail::PressureConditionTracking, 786
serif::operators::backend::mfem::CoefficientTracking, 351, 352
serif::utils::dependencies::ProjectedSnapshot< FunctionT >, 801
serif::utils::dependencies::ValueSnapshot, 999
Captured
serif::utils::dependencies::detail::SourceSnapshot< ResourceT >, 918
captured
serif::utils::dependencies::detail::SourceSnapshot< ResourceT >, 918
CarrierQuantity
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 794
CarrierTerm
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 794
center
serif::equations::RigidRotation, 883
center_of_mass
serif::discretization::quadrature::QuadratureOptions, 806
Centering
serif::operators::backend::mfem, 169
CentralDensity
serif::operators, 156
CentralDensityConstraint

serif::equations::CentralDensityConstraint< EOST >, 341
centrifugal
serif::discretization::quadrature::QuadratureOptions, 806
changed
serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >, 383
changed_inputs
serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >, 383
changed_sources
serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >, 783
check_independent_harmonics
serif::geometry::deformation::detail, 135
checked_data
serif::operators::backend::mfem::JacobianOperator< CacheT >, 659
checked_linearization
serif::geometry::surface::detail, 144
checked_mesh
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >, 967
checked_value
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >, 746
ChemicalPotentialValue
serif::dimensions, 62
clear
serif::discretization::backend::mfem::TensorBasisCache, 956
serif::operators::backend::mfem::TensorBasisCache, 958
clear_geometry_cache
serif::mapping::DomainMapper, 469
clone
serif::mapping::compactification::ExteriorDomainMap, 525
serif::mapping::compactification::KelvinCompactification, 663
serif::mapping::DomainMapper, 469
Closure
serif::operators, 156
serif::operators::backend::mfem, 169
coefficient
serif::equations::EquationScale< EquationT >, 521
serif::operators::ContributionBinding< EquationT >, 389
coefficient_order
serif::discretization::quadrature::Query, 820
coefficients
serif::discretization::backend::mfem::TensorWorkspace, 960
coefficients_term
serif::discretization::blocks::surface_deformation, 938
collect_volume_results
serif::operators::backend::mfem, 176
collection
serif::geometry::deformation::backend::mfem::ReferenceSpaceSna 837
column_count
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 515
column_index
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 514
Columns
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 513
columns
serif::operators::backend::mfem::BasisBindings< ColumnCount >, 291
CombinedKeyList
serif::utils::dependencies, 198
compactification
serif::mapping::ElementMappingData, 492
serif::operators::backend::mfem::ElementGeometry< OperandCount >, 491
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >, 1013
compactification_coordinate
serif::mapping::compactification::ExteriorMapInput, 527
serif::mapping::compactification::KelvinCompactification::Radial 826
serif::mapping::CompactificationPointData, 356
compactification_coordinate_gradient
serif::mapping::compactification::ExteriorMapInput, 527
COMPACTIFICATION_MISMATCH
serif::utils::errors, 209
compactification_sequence
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >, 1013
compactification_space
serif::operators::backend::mfem::BasisBindings< ColumnCount >, 291
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >, 1013
compactified
serif::mapping::MappingPointContext, 677
compatible_error_v
serif::utils::dependencies::detail, 206
compatible_error_v< Source, std::variant< Alternatives... > >

serif::utils::dependencies::detail, 206
compatible_errors
serif::utils::dependencies::detail, 206
compile_radial_reference
serif::geometry::deformation::backend::mfem::detail, 133
component
serif::discretization::forms, 95
component_count
serif::discretization::forms, 96
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 515
component_offset
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 514
components
serif::discretization::backend::mfem::TensorBasis, 952
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >, 716
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 732
compose_deformation
serif::geometry::deformation, 124
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 362
ComposedDeformation
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 360
Computation
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >, 364
computation
serif::utils::dependencies::ComputationGraph< Computations >, 377
computation_index
serif::utils::dependencies::PreparationFailure< ErrorT >, 779
computational_radius
serif::mapping::compactification::KelvinCompactification, 826
ComputationAt
serif::utils::dependencies::ComputationGraph< Computations >, 376
ComputationCache
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 368
ComputationGraph
serif::utils::dependencies::ComputationGraph< Computations >, 377
ComputationStore
serif::utils::dependencies::ComputationStore< EntryTs >, 381
compute_base_order
serif::discretization::quadrature::Policy, 765
compute_hdiv_mass_tensor
serif::mapping, 150
compute_hdiv_mass_tensor_variation
serif::mapping, 150
compute_radial_factors
serif::mapping::compactification::KelvinCompactification, 663
compute_scalar_diffusion_tensor
serif::mapping, 150
computed
serif::utils::dependencies, 199
computed_count
serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >, 782
ConcatT
serif::utils::misc::concepts, 213
configuration
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >, 1011
configure
serif::discretization::quadrature::backend::mfem::MFEMResolver, 707
connect_volume_contributions
serif::operators::backend::mfem, 177
connectedDiagnostics
serif::discretization::domain::relation::validation::RelationValidati 850
serif::discretization::domain::schema::validation::RelationValidati 852
Constant
serif::equations::BarotropicHydrostaticEquilibrium, 286
serif::operators, 156
serif::operators::BarotropicHydrostaticEquilibrium, 288
contains
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 368
contains_boundary
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 479
contains_domain
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 479
contains_id
serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > >, 325
serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > >, 326
serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList<

DomainIDTs... > >, 465
serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > >, 466
contains_key_v
serif::utils::dependencies::detail, 206
contains_key_v< KeyT, KeyList< Keys... > >
serif::utils::dependencies::detail, 207
continuation_identity
serif::geometry::surface::IsobaricSurface, 647
Continuity
serif::discretization::functional_space, 98
continuity
serif::discretization::functional_space::SpaceDescriptor, 919
Continuity2StringMap
serif::discretization::functional_space, 99
continuous
serif::discretization::functional_space, 98
contract_axis
serif::discretization::backend::mfem::TensorBasis, 953
contraction_mode
serif::operators::backend::mfem::VolumeOptions, 1022
ContractionMode
serif::discretization::backend::mfem, 70
serif::operators::backend::mfem, 172
contravariant_piola
serif::discretization::functional_space, 99
ContributionChildren
serif::operators::detail::ContributionChildren< EquationT >, 390
contributions
serif::equations::DisplacementForce< EOST >, 427
serif::equations::EquationDifference< LeftT, RightT >, 511
serif::equations::EquationScale< EquationT >, 521
serif::equations::EquationSum< FirstT, RestTs >, 523
serif::equations::GravityGradientRelation, 615
serif::equations::PoissonBalance, 764
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >, 1011
serif::operators::BadContributionCollection, 278
serif::operators::BadContributionRow, 280
serif::operators::NestedBadCollection, 713
ContributionsOf
serif::equations, 117
control
serif::discretization::quadrature::FormRule, 573
convergence
serif::discretization::quadrature, 104
convergence_boosts
serif::discretization::quadrature::QuadratureOptions, 806
coordinate_gradient
serif::mapping::CompactificationPointData, 356
coordinate_tolerance
serif::utils::user_options::KelvinCompactificationOptions, 665
copies
serif::discretization::backend::mfem::TensorBasisCache::Entry, 500
serif::operators::backend::mfem::TensorBasisCache::Entry, 501
CoreEnvelopeVacuumDomainSchema
serif::discretization::domain::schema, 89
count
serif::discretization::domain::BoundarySet< BoundaryTs >, 327
serif::discretization::domain::DomainOrSetList< DomainTs >, 476
serif::discretization::domain::DomainSet< DomainTs >, 481
serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs >, 323
serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >, 463
serif::discretization::domain::relation::lists::RelationList< RelationTs >, 844
serif::discretization::domain::schema::BoundaryIDList< BoundaryIDTs >, 324
serif::discretization::domain::schema::DomainIDList< DomainIDTs >, 464
serif::discretization::domain::schema::RelationList< RelationTs >, 844
serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >, 738
serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Relations... > >, 696
CountVariadicArguments
serif::discretization::domain, 78
serif::discretization::domain::ids::lists, 82
serif::discretization::domain::relation, 85
serif::discretization::domain::relation::lists, 86
serif::eos::relations, 112
serif::utils::misc::std_helper, 215
covariant_piola
serif::discretization::functional_space, 99
ct_error_string
serif::utils::types::ct_error_string< N >, 400
ctstring
serif::utils::types::ctstring< N >, 401
CYAN

serif::utils::misc::terminal, 217

data
serif::utils::types::ct_error_string< N >, 400
serif::utils::types::ctstring< N >, 401
deformation_discretization
serif::geometry::deformation::backend::mfem, 129
DeformationFailureReason
serif::geometry::deformation, 122
DeformationStatus
serif::geometry::deformation, 122
DeformationWorkspace
serif::geometry::deformation, 122
degree
serif::geometry::deformation::SphericalHarmonicMode, 933
dense
serif::discretization::backend::mfem, 70
Density
serif::equations::AngularMomentum, 266
serif::equations::gravity_contributions, 118
serif::equations::GravityForce, 612
serif::equations::MassNormalization, 694
serif::equations::RotationalForce, 889
serif::operators, 157
serif::operators::backend::mfem, 170
density_from_enthalpy
serif::equations::CentralDensityConstraint< EOST >, 341
density_projection
serif::discretization::quadrature::QuadratureOptions, 806
DensityBinding
serif::operators::backend::mfem, 170
DensityFromSpecificEnthalpy
serif::eos::relations, 111
DensitySource
serif::equations::gravity_contributions::DensitySource, 412
DensityValue
serif::dimensions, 62
serif::equations::AngularMomentum, 266
serif::equations::GravityForce, 612
serif::equations::MassNormalization, 694
serif::equations::RotationalForce, 889
serif::operators, 157
Dependencies
serif::utils::dependencies::ComputationGraph< Computations >, 376
DependencyContext
serif::utils::dependencies::DependencyContext< ResourceTs >, 414
DependencyContextView
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >, 417
derivative
serif::operators::backend::mfem::NodalData< EquationT, SchemaT >, 714
derivative_component
serif::discretization::forms, 95
descriptor
serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >, 905
serif::geometry::surface::IsobaricSurface, 647
descriptors
serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs >, 323
serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >, 463
serif::discretization::domain::schema::BoundaryIDList< BoundaryIDTs >, 324
serif::discretization::domain::schema::DomainIDList< DomainIDTs >, 464
serif::eos::runtime::RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > >, 898
determinant_floor
serif::geometry::deformation::GeometryStepOptions, 594
determinant_polynomial
serif::geometry::deformation::detail, 135
DeterminantPolynomial
serif::geometry::deformation::detail, 135
diagnostic
serif::discretization::quadrature, 104
serif::discretization::quadrature::QuadratureRoleOptions, 810
serif::discretization::quadrature::RoleControls, 885
did_work
serif::operators::backend::mfem::NumericalWork, 724
dimension
serif::utils::user_options::DomainMapperOptions, 474
DimensionlessValue
serif::dimensions, 62
DISABLE_WARNING
warning.hpp, 1369
discontinuous
serif::discretization::functional_space, 98
Discretization
serif::operators::backend::mfem::NodalData< EquationT, SchemaT >, 714
discretization
serif::discretization::quadrature, 104

serif::discretization::quadrature::QuadratureRoleOptions, 810
serif::discretization::quadrature::RoleControls, 885
serif::operators::backend::mfem::NodalData< EquationT, SchemaT >, 714
serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >, 1008
displaced_position
serif::mapping::compactification::ExteriorMapInput, 527
serif::mapping::MappingPointContext, 677
displaced_position_variation
serif::mapping::compactification::ExteriorMapDirection, 526
Displacement
serif::operators, 157
displacement
serif::mapping::ElementMappingData, 492
serif::operators::backend::mfem::ElementGeometry< OperandCount >, 491
displacement_basis
serif::operators::backend::mfem::ElementBasis< OperandCount >, 484
displacement_binding
serif::operators::backend::mfem::BasisBindings< ColumnCount >, 291
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >, 1013
displacement_direction
serif::operators::backend::mfem::Workspace, 1027
displacement_dual
serif::operators::backend::mfem::Workspace, 1027
displacement_gradient
serif::operators::backend::mfem::Workspace, 1027
displacement_jacobian
serif::mapping::compactification::ExteriorMapInput, 527
serif::mapping::MappingPointContext, 677
displacement_jacobian_variation
serif::mapping::compactification::ExteriorMapDirection, 526
serif::mapping::MappingPointVariation, 678
displacement_samples
serif::operators::backend::mfem::Workspace, 1027
displacement_space
serif::geometry::deformation::backend::mfem::ReferenceDiscretization, 831
displacement_value
serif::operators::backend::mfem::Workspace, 1027
displacement_variation
serif::mapping::MappingPointVariation, 678
DisplacementBinding
serif::operators::backend::mfem, 170
serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >, 1007
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >, 1013
DisplacementDivergence
serif::operators, 157
DisplacementForce
serif::equations::DisplacementForce< EOST >, 427
DisplacementValue
serif::operators, 157
Divergence
serif::equations::gravity_contributions, 118
divergence
serif::discretization::backend::mfem, 70
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 732
DMWorkspace
serif::mapping::DMWorkspace, 435
docs/static/mainpage.md, 1031
DofMap
serif::discretization::backend::mfem::DofMap, 440
dofs
serif::discretization::backend::mfem::TensorBasis, 953
serif::discretization::backend::mfem::TensorBasis::Product, 799
DofSelection
serif::discretization::backend::mfem::DofSelection, 446
domain
serif::discretization::quadrature::backend::mfem::MFEMQuadratu 705
serif::discretization::quadrature::Query, 821
domain_count
serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >, 451
serif::discretization::domain::schema::validation::DomainBoundar BoundaryT, DomainTs >, 453
domain_descriptors
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 479
domain_id
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 479
domain_id_belongs_to
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 480

domain_id_count
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 480
domain_ids_type
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 478
domain_type
serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >, 461
serif::discretization::domain::relation::FullyConnected< A >, 585
serif::discretization::domain::schema::DomainID< DomainT, IdentifierValue >, 462
serif::discretization::domain::schema::validation::FullyConnected< A >, 587
domain_types
serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >, 451
serif::discretization::domain::schema::validation::DomainBoundary< BoundaryT, DomainTs >, 453
DomainAbsent
serif::discretization::domain::relation::validation, 87
DomainBoundaryAbsent
serif::discretization::domain::relation::validation, 87
domainBoundaryDiagnostics
serif::discretization::domain::relation::validation::RelationValidationResult, 850
serif::discretization::domain::schema::validation::RelationValidationResult, 852
DomainBoundaryExpectedFaceHasWrongID
serif::discretization::domain::relation::validation, 87
DomainBoundaryExpectedFaceIsUntagged
serif::discretization::domain::relation::validation, 87
DomainBoundaryTaggedFaceHasWrongTopology
serif::discretization::domain::relation::validation, 87
DomainBoundaryTaggedFaceTouchesUnexpectedDomain
serif::discretization::domain::relation::validation, 87
DomainDisconnected
serif::discretization::domain::relation::validation, 87
domainElementCount
serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics, 386
serif::discretization::domain::schema::validation::RelationValidationResult::ConnectedDiagnostics, 387
DomainMapper
serif::mapping::DomainMapper, 469
DomainOrSet
serif::discretization::domain::DomainOrSet, 475
domains
serif::discretization::domain::DomainOrSet, 476
ds
serif::mapping::FaceQuadratureContext, 532
DYNAMIC_BLOCK_SIZE
serif::discretization::blocks, 76
dynamic_orders
serif::discretization::quadrature::backend::mfem::MFEMQuadrat 705
DynamicViscosityValue
serif::dimensions, 62

edges
serif::utils::dependencies::detail::Graph< Computations >, 607
element
serif::discretization::backend::mfem::TensorBasisCache::Entry, 500
serif::geometry::deformation::backend::mfem::GeometrySampleR 592
serif::geometry::deformation::DeformationFailure, 403
serif::operators::backend::mfem::ElementBasis< OperandCount >, 484
serif::operators::backend::mfem::ElementData< OperandCount >, 487
serif::operators::backend::mfem::PreparationRejection, 781
serif::operators::backend::mfem::TensorBasisCache::Entry, 501
ELEMENT_1
serif::mapping, 150
ELEMENT_2
serif::mapping, 150
element_attributes
serif::geometry::deformation::backend::mfem::ReferenceMeshSna 836
element_count
serif::discretization::domain::mesh::MeshTopology, 700
element_displacement_data_from_element_vdofs
serif::mapping, 151
element_domain_id
serif::discretization::domain::mesh::MeshTopology, 700
element_equation_point
serif::operators::backend::mfem, 177
element_jacobian_action
serif::operators::backend::mfem, 177
element_jacobian_transpose
serif::operators::backend::mfem, 177
element_neighbors

serif::discretization::domain::mesh::MeshTopology, 700
ELEMENT_NULLPTR_ERROR
serif::utils::errors, 210
element_residual
serif::operators::backend::mfem, 178
element_result
serif::operators::backend::mfem::Workspace, 1027
element_support
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >, 967
element_values
serif::discretization::backend::mfem::DofMap, 441
serif::operators::backend::mfem, 178
serif::operators::backend::mfem::Workspace, 1028
ElementBatch
serif::operators::backend::mfem, 170
ElementCompactificationData
serif::mapping::ElementCompactificationData, 485
ElementDisplacementData
serif::mapping::ElementDisplacementData, 489
elementID
serif::discretization::domain::relation::validation::RelationValidationResult::ConnectedDiagnostics, 386
serif::discretization::domain::schema::validation::RelationValidationResult::ConnectedDiagnostics, 387
elements
serif::operators::backend::mfem::ContributionData< OperandCount >, 391
empty_surface
serif::geometry::surface, 139
enabled
serif::discretization::quadrature::QuadratureManifoldOptions, 804
serif::mapping::DMGeometryCache, 432
end
serif::discretization::quadrature::RuleSet, 892, 893
EnergyDensityValue
serif::dimensions, 63
EnergyValue
serif::dimensions, 63
Enthalpy
serif::equations::BarotropicClosure< EOST >, 283
serif::equations::PressureForce< EOST >, 788
serif::equations::PressureSurfaceConstraint< EOST >, 791
serif::operators, 157
enthalpy_scale
serif::eos::models::Polytrope, 768
EnthalpyBinding
serif::operators::backend::mfem, 170
EnthalpyGradient
serif::operators, 157
EnthalpyValue
serif::equations::PressureForce< EOST >, 788
serif::operators, 157
EntropyValue
serif::dimensions, 63
Entry
serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >, 737
enum_to_enum_string_map
serif::meta, 154
enum_to_string
serif::meta, 154
enum_to_string_enum_map
serif::meta, 154
EnumerateTypesT
serif::utils::misc::concepts, 213
eos_closure
serif::discretization::quadrature::QuadratureOptions, 806
EOS_ERROR
serif::utils::errors, 209
EOS_INVALID_THERMODYNAMIC_INPUT
serif::utils::errors, 209
EOS_NONFINITE_INPUT
serif::utils::errors, 209
EOS_NONFINITE_RESULT
serif::utils::errors, 209
EOS_OUTSIDE_DOMAIN
serif::utils::errors, 209
EOS_UNSUPPORTED_DERIVATIVE
serif::utils::errors, 209
EOS_UNSUPPORTED_RELATION
serif::utils::errors, 209
EOS_WRONG_INPUT_COUNT
serif::utils::errors, 209
EOS_WRONG_INPUT_QUANTITY
serif::utils::errors, 209
EOSEvaluationError
serif::eos::EOSEvaluationError, 505
EOSStateQuantities
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 794
EOSStateTerms
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 794
EOSView
serif::eos::runtime::EOSView, 507

EQUATION
serif::operators::backend::mfem, 173
Equation
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >, 715
serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >, 761
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >, 1011
equation
serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >, 1008
serif::operators::ContributionBinding< EquationT >, 389
equation_rule
serif::operators::backend::mfem, 178
EquationDifference
serif::equations::EquationDifference< LeftT, RightT >, 511
EquationOfState
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 794
EquationRegistry
serif::equations::EquationRegistry< EquationTs >, 517
serif::operators::EquationRegistry< EquationTs >, 519
Equations
serif::equations::EquationRegistry< EquationTs >, 517
serif::operators::EquationRegistry< EquationTs >, 518
EquationScale
serif::equations::EquationScale< EquationT >, 521
EquationSum
serif::equations::EquationSum< FirstT, RestTs >, 523
Error
serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >, 880
serif::utils::dependencies::detail::Result< T >, 879
error
serif::utils::dependencies::PreparationFailure< ErrorT >, 779
error_norm
serif::discretization::quadrature::QuadratureOptions, 806
estimate_geometry_step_limit
serif::geometry::deformation::backend::mfem, 129
Eval
evaluate
serif::discretization::backend::mfem::BasisEvaluation, 296
serif::eos, 107
serif::eos::models::Polytrope, 768
serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > >, 895
serif::equations::AngularMomentum, 267
serif::equations::BarotropicClosure< EOST >, 284
serif::equations::BarotropicHydrostaticEquilibrium, 287
serif::equations::CenteringConstraint, 337
serif::equations::CentralDensityConstraint< EOST >, 341
serif::equations::gravity_contributions::DensitySource, 412
serif::equations::gravity_contributions::GradientDivergence, 600
serif::equations::gravity_contributions::GradientValue, 604
serif::equations::gravity_contributions::PotentialValue, 775
serif::equations::GravityForce, 613
serif::equations::MassNormalization, 694
serif::equations::PressureForce< EOST >, 789
serif::equations::PressureSurfaceConstraint< EOST >, 792
serif::equations::RotationalForce, 890
serif::mapping::compactification::ExteriorDomainMap, 525
serif::mapping::compactification::KelvinCompactification, 663
serif::operators::backend::mfem::BasisEvaluation, 303
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 730
serif::operators::BarotropicHydrostaticEquilibrium, 289
serif::operators::GradientRequest, 602
serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >, 719
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >, 364
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >, 631
evaluate_compactification_coordinate
serif::mapping::DomainMapper, 470
evaluate_contribution_equation
serif::operators::backend::mfem, 178
evaluate_displacement
serif::mapping::PhysicalPositionFunctionCoefficient, 757

serif::geometry::deformation, 125
evaluate_displacement_field
serif::mapping::DomainMapper, 470
evaluate_element_equation
serif::operators::backend::mfem, 179
evaluate_face
serif::mapping::DomainMapper, 470
serif::mapping::GridFunctionMappingEvaluator, 616
evaluate_face_variation
serif::mapping::DomainMapper, 470
evaluate_jacobian_variation_factors
serif::mapping::compactification::KelvinCompactification, 663
evaluate_nodal_values
serif::operators::backend::mfem, 179
evaluate_point
serif::mapping::DomainMapper, 470
serif::mapping::GridFunctionMappingEvaluator, 616
evaluate_point_impl
serif::mapping::DomainMapper, 471
evaluate_point_variation
serif::mapping::DomainMapper, 471
evaluate_point_variation_impl
serif::mapping::DomainMapper, 471
evaluate_polynomial
serif::geometry::deformation::detail, 135
evaluate_pressure_surface
serif::geometry::surface, 139
evaluate_relation
serif::geometry::surface::detail, 144
evaluate_response_samples
serif::geometry::surface::detail, 144
evaluate_surface_response
serif::geometry::surface::detail, 144
evaluate_update
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >, 631
evaluate_variation
serif::mapping::compactification::ExteriorDomainMap, 525
serif::mapping::compactification::KelvinCompactification, 663
evaluate_volume
serif::mapping::DomainMapper, 471
serif::mapping::GridFunctionMappingEvaluator, 617
evaluate_volume_variation
serif::mapping::DomainMapper, 472
evaluateRuntimePartialDerivative
serif::eos::runtime, 114
evaluateRuntimeRelation
serif::eos::runtime, 114
EVALUATOR_ERROR
serif::utils::errors, 209
expectedBoundaryID
serif::discretization::domain::relation::validation::RelationValidati 454
serif::discretization::domain::schema::validation::RelationValidati 456
extension_map
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 361
extension_workspace
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace, 1025

face_count
serif::discretization::domain::mesh::MeshTopology, 700
face_elements
serif::discretization::domain::mesh::MeshTopology, 700
FaceElementSide
serif::mapping, 150
faceID
serif::discretization::domain::relation::validation::RelationValidati 454
serif::discretization::domain::relation::validation::RelationValidati 637
serif::discretization::domain::schema::validation::RelationValidati 456
serif::discretization::domain::schema::validation::RelationValidati 638
factors
serif::discretization::backend::mfem::TensorBasis::Product, 799
failed_relation_count
serif::discretization::domain::schema::validation::SchemaValidatio 909
FAILED_TYPE_INVARIANT
serif::utils::errors, 210
Failure
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 367
failure
serif::discretization::domain::relation::validation::RelationValidati 850
serif::discretization::domain::schema::validation::RelationValidati 852
fallback_control
serif::discretization::quadrature::RuleSet, 893
fallback_fixed_order
serif::discretization::quadrature::QuadratureOptions, 807
fallible

serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >, 881
serif::utils::dependencies::detail::Result< T >, 880
Family
serif::discretization::functional_space, 98
family
serif::discretization::blocks::SpatialTermTraits< TermT >, 922
serif::discretization::functional_space::SpaceDescriptor, 919
Family2StringMap
serif::discretization::functional_space, 99
fast
serif::discretization::quadrature, 104
field
serif::discretization::blocks::TermDescriptor, 969
field_derivatives
serif::equations::PointLinearization< OperandsT, ResultT >, 762
serif::geometry::surface::SurfaceEOSResponse< Formulation >, 945
field_descriptor
serif::discretization::blocks, 76
field_type
serif::discretization::blocks::term< FieldT >, 962
FieldOf
serif::discretization::blocks, 75
fields
serif::equations::EquationPoint< OperandsT >, 516
FilteredTail
serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate >, 536
FilterT
serif::utils::misc::concepts, 213
find_form_control
serif::discretization::quadrature::RuleSet, 893
find_relation
serif::eos::runtime::EOSView, 508
finite_state
serif::geometry::surface::detail, 145
first
serif::discretization::backend::mfem::TensorWorkspace, 960
first_boundary_parameter
serif::geometry::deformation::detail, 136
first_failed_relation_index
serif::discretization::domain::schema::validation::SchemaValidationResult, 909
firstDomainID
serif::discretization::domain::relation::validation::RelationValidationResult::DomainBoundaryDiagnostics, 454
serif::discretization::domain::schema::validation::RelationValidationResult::DomainBoundaryDiagnostics, 456
firstElementID
serif::discretization::domain::mesh::FaceElements, 529
serif::discretization::domain::relation::validation::RelationValidati
454
serif::discretization::domain::schema::validation::RelationValidati
456
fixed_order
serif::discretization::quadrature::QuadratureTermOptions, 810
serif::discretization::quadrature::RuleControl, 891
for_each
serif::equations::EquationRegistry< EquationTs >, 517
serif::operators::EquationRegistry< EquationTs >, 519
for_each_column
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 514
for_each_contribution
serif::operators, 159
for_each_input
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 514
for_each_operand
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 514
for_each_volume_element
serif::operators::backend::mfem, 179
ForceValue
serif::dimensions, 63
Form
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 513
form
serif::discretization::quadrature::FormRule, 573
serif::discretization::quadrature::Query, 821
form_descriptor
serif::discretization::forms, 96
format
std::formatter< serif::discretization::blocks::FieldDescriptor >, 544
std::formatter< serif::discretization::blocks::TermDescriptor >, 545
std::formatter< serif::discretization::domain::DomainOrSet >, 547
std::formatter< serif::discretization::domain::schema::validation::
>, 547
std::formatter< serif::discretization::forms::FormDescriptor >, 549
std::formatter< serif::discretization::functional_space::Continuity >, 551

std::formatter< serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R > >, 552
std::formatter< serif::discretization::functional_space::FamilyKind >, 554
std::formatter< serif::discretization::functional_space::MappingKind >, 555
std::formatter< serif::discretization::functional_space::RankKind R > >, 557
std::formatter< serif::discretization::functional_space::SpaceDescriptor >, 558
std::formatter< serif::discretization::functional_space::TraceKind >, 560
std::formatter< serif::discretization::quadrature::backend::mfem::MFEMQuadrature >, 561
std::formatter< serif::discretization::quadrature::FormRank >, 562
std::formatter< serif::discretization::quadrature::MappingKind >, 564
std::formatter< serif::discretization::quadrature::Mode >, 565
std::formatter< serif::discretization::quadrature::QuadratureRole >, 566
std::formatter< serif::discretization::quadrature::Query >, 568
std::formatter< serif::discretization::quadrature::RuleControl >, 569
std::formatter< serif::discretization::quadrature::RuleSet >, 570
four_pi_G
serif::equations::gravity_contributions::DensitySource, 412
fraction_to_boundary
serif::geometry::deformation::GeometryStepOptions, 594
FrequencyValue
serif::dimensions, 63
full_result
serif::operators::backend::mfem::Workspace, 1028
full_size
serif::discretization::backend::mfem::DofMap, 441
Func
serif::mapping::PhysicalPositionFunctionCoefficient, 757
Function
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >, 363
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >, 630
functional_space
serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >, 422
FunctionalSpace
serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >, 422
serif::discretization::functional_space::density::mass, 683
serif::discretization::blocks::displacement::vector, 1002
serif::discretization::blocks::enthalpy::specific, 925
serif::discretization::blocks::gravity::gradient, 598
serif::discretization::blocks::gravity::potential, 772
serif::discretization::blocks::SpatialTermTraits< TermT >, 922
FunctionalSpaceOf
serif::discretization::backend::mfem::MFEMQuadrature, 75
FunctionalSpaceTraits
serif::discretization::quadrature::backend::mfem, 105
gather
serif::discretization::backend::mfem::DofMap, 441
general
serif::discretization::quadrature, 103
GeneratedType
serif::discretization::blocks::generated_residual_block< GeneratedResidual >, 589
serif::discretization::blocks::generated_value_block< GeneratedValue >, 591
geometry
serif::discretization::quadrature::backend::mfem::MFEMQuadrature, 706
serif::operators::backend::mfem::ElementData< OperandCount >, 487
geometry_dependent
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 732
geometry_integrand_variation
serif::operators::backend::mfem, 179
geometry_points
serif::operators::backend::mfem::NumericalWork, 724
geometry_variation
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 730
geometry_weight_order
serif::discretization::quadrature::backend::mfem::MFEMQuadrature, 706
serif::discretization::quadrature::Query, 821
get
serif::discretization::backend::mfem::TensorBasisCache, 956
serif::equations::EquationRegistry< EquationTs >, 517
serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >, 738

serif::operators::backend::mfem::TensorBasisCache, 958
serif::operators::EquationRegistry< EquationTs >, 519
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 368
serif::utils::dependencies::ComputationStore< EntryTs >, 381, 382
serif::utils::dependencies::DependencyContext< ResourceTs >, 414
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >, 417
serif::utils::dependencies::detail::BorrowedValue< T >, 315
serif::utils::dependencies::detail::OwnedValue< T >, 748
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >, 746, 747
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >, 877
get_coordinate_tolerance
serif::mapping::compactification::KelvinCompactification, 663
get_dimension
serif::mapping::DMWorkspace, 435
serif::mapping::DomainMapper, 472
serif::mapping::ElementDisplacementData, 489
serif::mapping::VolumeMappingCache, 1015
get_dof_count
serif::mapping::ElementCompactificationData, 485
serif::mapping::ElementDisplacementData, 489
get_dof_matrix
serif::mapping::ElementDisplacementData, 489
get_dofs
serif::mapping::ElementCompactificationData, 485
get_element
serif::mapping::ElementCompactificationData, 486
serif::mapping::ElementDisplacementData, 489
get_exterior_map
serif::mapping::DomainMapper, 472
get_face_quadrature_context
serif::mapping::GridFunctionMappingEvaluator, 617
get_fixed_mesh_geometry
serif::mapping::DomainMapper, 472
get_geometry_cache_statistics
serif::mapping::DomainMapper, 472
get_name
serif::mapping::compactification::ExteriorDomainMap, 525
serif::mapping::compactification::KelvinCompactification, 663
get_ordering
serif::mapping::ElementDisplacementData, 489
get_physical_point
serif::mapping::GridFunctionMappingEvaluator, 617
get_point_count
serif::mapping::VolumeMappingCache, 1015
get_point_data
serif::mapping::VolumeMappingCache, 1016
get_quadrature_context
serif::mapping::GridFunctionMappingEvaluator, 617
get_reference_infinity_radius
serif::mapping::compactification::KelvinCompactification, 664
get_reference_stellar_radius
serif::mapping::compactification::KelvinCompactification, 664
get_role_control
serif::discretization::quadrature::Policy, 765
global_boost
serif::discretization::quadrature::QuadratureOptions, 807
global_scalar
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 732
Gradient
serif::equations::gravity_contributions, 118
serif::equations::GravityForce, 612
gradient
serif::discretization::backend::mfem, 70
serif::equations::RigidRotation, 884
serif::mapping::DMReferenceDisplacementSample, 433
gradient_term
serif::discretization::blocks::gravity, 611
gradient_trace
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 730
GradientValue
serif::equations::GravityForce, 613
gravitational_energy
serif::discretization::quadrature::QuadratureOptions, 807
GravitationalPotentialValue
serif::dimensions, 63
gravity_boundary
serif::discretization::quadrature::QuadratureOptions, 807
gravity_divergence
serif::discretization::quadrature::QuadratureOptions, 807
gravity_force

serif::discretization::quadrature::QuadratureOptions, 807
gravity_hdiv_mass
serif::discretization::quadrature::QuadratureOptions, 807
gravity_source
serif::discretization::quadrature::QuadratureOptions, 807
GravityDivergence
serif::operators, 157
GravityValue
serif::operators, 157
GREEN
serif::utils::misc::terminal, 217
GridFunctionMappingEvaluator
serif::mapping::GridFunctionMappingEvaluator, 616

H1
serif::discretization::functional_space, 99
h1
serif::discretization::functional_space, 98
h1_divergence
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 732
has_binding_v
serif::utils::dependencies, 201
has_nodal_parameters
serif::geometry::deformation::SurfaceEmbedding, 941
hasPartialDerivative
serif::dimensions::runtime::RuntimeRelationDescriptor, 900
Hcurl
serif::discretization::functional_space, 100
hcurl
serif::discretization::functional_space, 98
Hdiv
serif::discretization::functional_space, 100
hdiv
serif::discretization::functional_space, 98
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 732
height
serif::discretization::backend::mfem::BasisEvaluation, 296
serif::operators::backend::mfem::BasisEvaluation, 303
hessian
serif::equations::RigidRotation, 884
hits
serif::mapping::DMGeometryCache, 432
serif::mapping::DMGeometryCacheStatistics, 433
hydrostatic_equilibrium
serif::discretization::quadrature::QuadratureOptions, 807

ID
serif::discretization::domain::ids::BoundaryDescriptor, 319
serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >, 320
serif::discretization::domain::ids::DomainDescriptor, 459
serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >, 461
serif::discretization::domain::schema::BoundaryID< BoundaryT, IdentifierValue >, 322
serif::discretization::domain::schema::DomainID< DomainT, IdentifierValue >, 462
id
serif::dimensions::runtime::RuntimeQuantityValue, 900
serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > >, 326
serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > >, 466
identifier
serif::dimensions::Acceleration, 260
serif::dimensions::AngularMomentum, 264
serif::dimensions::AngularVelocity, 270
serif::dimensions::Area, 274
serif::dimensions::ChemicalPotential, 344
serif::dimensions::Dimensionless, 419
serif::dimensions::DynamicViscosity, 483
serif::dimensions::Energy, 494
serif::dimensions::EnergyDensity, 495
serif::dimensions::Entropy, 499
serif::dimensions::Force, 539
serif::dimensions::Frequency, 584
serif::dimensions::GravitationalPotential, 609
serif::dimensions::InternalEnergy, 640
serif::dimensions::KinematicViscosity, 670
serif::dimensions::Length, 671
serif::dimensions::Luminosity, 674
serif::dimensions::MagneticFluxDensity, 676
serif::dimensions::Mass, 680
serif::dimensions::MassDensity, 691
serif::dimensions::MassFlowRate, 692
serif::dimensions::MomentOfInertia, 710
serif::dimensions::Momentum, 711
serif::dimensions::NumberDensity, 723
serif::dimensions::Opacity, 726
serif::dimensions::Power, 777
serif::dimensions::Pressure, 785

serif::dimensions::SpecificEnergy, 927
serif::dimensions::SpecificEnthalpy, 929
serif::dimensions::SpecificEntropy, 930
serif::dimensions::SpecificInternalEnergy, 932
serif::dimensions::Temperature, 949
serif::dimensions::Time, 976
serif::dimensions::Torque, 978
serif::dimensions::Velocity, 1005
serif::dimensions::Volume, 1006
identity
serif::discretization::backend::mfem::CoefficientTracking, 349
serif::discretization::functional_space, 99
serif::geometry::deformation::backend::mfem::DeformationValueTracking, 406
serif::geometry::deformation::backend::mfem::ReferenceDiscretization, 834
serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot, 836
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot, 837
serif::geometry::surface::detail::BoundaryValueTracking, 330
serif::geometry::surface::detail::PressureConditionTracking, 786
serif::operators::backend::mfem::CoefficientTracking, 352
serif::operators::backend::mfem::MetadataTracking, 703
serif::operators::backend::mfem::NodalValueTracking, 721
serif::utils::dependencies::detail::SourceSnapshot< ResourceT >, 918
IMPROPERLY_ORDERED_REFERENCE_RADII
serif::utils::errors, 209
include_repeated_queries
serif::discretization::quadrature::QuadratureManifestOptions, 804
incompatible_discretization
serif::geometry::deformation, 122
inconsistent_sample_count
serif::geometry::surface, 139
IncrementalComputation
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >, 630
independent_results
serif::utils::dependencies::detail, 206
index
serif::discretization::blocks::residual_block< index_value >, 867
serif::discretization::blocks::value_block< index_value >, 996
serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >, 738
serif::operators::backend::mfem, 191
serif::utils::dependencies::ComputationGraph< Computations >, 377
infinity_attribute
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza 831
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza 833
infinity_radius
serif::geometry::deformation::RadialInterpolation, 828
inner_type
serif::discretization::domain::relation::Inscribed< AuxBracking, 634
serif::discretization::domain::schema::validation::Inscribed< AuxBracking, 636
InnerDomainAbsent
serif::discretization::domain::relation::validation, 87
InnerDomainTouchesMeshBoundary
serif::discretization::domain::relation::validation, 87
InnerDomainTouchesMeshBoundary
serif::discretization::domain::relation::validation, 87
InnerDomainTouchesUnexpectedDomain
serif::discretization::domain::relation::validation, 87
innerElementID
serif::discretization::domain::relation::validation::RelationValidati 637
serif::discretization::domain::schema::validation::RelationValidati 638
input
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 369
input_size
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 361
serif::geometry::deformation::RadialExtension, 824
serif::geometry::deformation::SurfaceEmbedding, 941
input_versions
serif::utils::dependencies::detail::CachedResult< ValueT, InputCount >, 333
inputCount
serif::eos::relations::Relation< Output, Inputs >, 841
InputList
serif::utils::dependencies, 198
InputQuantities
serif::eos::relations::Relation< Output, Inputs >, 840

inputQuantities
serif::dimensions::runtime::RuntimeRelationDescription, 901
inputQuantityIDs
serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >, 905
Inputs
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >, 363
serif::utils::dependencies::ComputationGraph< Computations >, 377
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >, 630
InputSpace
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 360
serif::geometry::deformation::RadialExtension, 823
serif::geometry::deformation::SurfaceEmbedding, 941
inscribedDiagnostics
serif::discretization::domain::relation::validation::RelationValidationResult, 851
serif::discretization::domain::schema::validation::RelationValidationResult, 852
inspect_source
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 369
integrate
serif::discretization::backend::mfem::BasisEvaluation, 296
serif::operators::backend::mfem::BasisEvaluation, 303
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 730
integration_order
serif::operators::backend::mfem, 180
integration_rule
serif::discretization::quadrature::backend::mfem::MFEMRule, 708
serif::geometry::deformation::backend::mfem::GeometryIntegrationRule, 592
InternalEnergyValue
serif::dimensions, 63
interpolation
serif::geometry::deformation::RadialExtension, 824
interpolation_weights
serif::geometry::deformation::RadialInterpolation, 828
INVALID_ARGUMENT
serif::utils::errors, 209
INVALID_COMPACTIFICATION_DERIVATIVE_TYPE
serif::utils::errors, 209
INVALID_COMPACTIFICATION_RANK
serif::utils::errors, 209
INVALID_COMPACTIFICATION_TYPE
serif::utils::errors, 209
invalid_configuration
serif::geometry::deformation, 122
INVALID_COORDINATE_TOLERANCE
serif::utils::errors, 209
INVALID_DIMENSION
serif::mapping, 150
serif::utils::errors, 209
INVALID_DOF_COUNT
serif::utils::errors, 209
INVALID_DOF_ORDERING
serif::utils::errors, 209
INVALID_DOMAIN_ERROR
serif::utils::errors, 209
INVALID_ELEMENT_GEOMETRY
serif::utils::errors, 209
INVALID_ELEMENT_ID_ERROR
serif::utils::errors, 210
invalid_geometry
serif::geometry::deformation, 122
INVALID_MAPPING
serif::utils::errors, 209
INVALID_QUADRATURE_ORDER
serif::utils::errors, 209
INVALID_REFERENCE_RADIUS
serif::mapping, 150
serif::utils::errors, 209
invalid_size
serif::geometry::deformation, 122
INVALID_VACUUM_ELEMENT_ATTRIBUTE_ID
serif::utils::errors, 209
invalidate
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 369
invalidate_cache
serif::mapping::GridFunctionMappingEvaluator, 617
invalidate_changed_spaces
serif::mapping::GridFunctionMappingEvaluator, 617
inverse_element_jacobian_variation
serif::mapping::VolumeMappingVariation, 1019
inverse_jacobian
serif::mapping::DMFixedMeshGeometry, 431
inverse_mapping_jacobian
serif::mapping::MappingPointContext, 677
inverse_mapping_jacobian_variation
serif::mapping::MappingPointVariation, 678
INVERTED_JACOBIAN
serif::utils::errors, 209

is_affine_in_displacement
serif::mapping::compactification::ExteriorDomainMap, 525
serif::mapping::compactification::KelvinCompactification, 664
is_bindings_v
serif::utils::dependencies, 201
is_bindings_v< Bindings< BindingTs... > >
serif::utils::dependencies, 201
is_boundary_id_list_v
serif::discretization::domain::ids::lists, 82
is_boundary_id_list_v< BoundaryIDList< BoundaryIDTs... > >
serif::discretization::domain::ids::lists, 82
is_boundary_id_v
serif::discretization::domain::ids, 80
is_boundary_id_v< BoundaryID< BoundaryT, IdentifierValue > >
serif::discretization::domain::ids, 80
is_compactified_element
serif::mapping::DomainMapper, 472
is_computation_graph_v
serif::utils::dependencies, 202
is_computation_graph_v< ComputationGraph< Computations... > >
serif::utils::dependencies, 202
is_computation_v
serif::utils::dependencies, 202
is_computation_v< Computation< KeyT, InputsT, FunctionT > >
serif::utils::dependencies, 202
is_computation_v< IncrementalComputation< KeyT, ValueT, InputsT, FunctionT > >
serif::utils::dependencies, 202
is_dependency_context_v
serif::utils::dependencies, 202
is_dependency_context_v< DependencyContext< Resources... > >
serif::utils::dependencies, 202
is_dependency_context_v< DependencyContextView< ParentT, KeysT > >
serif::utils::dependencies, 202
is_domain_id_list_v
serif::discretization::domain::ids::lists, 83
is_domain_id_list_v< DomainIDList< DomainIDTs... > >
serif::discretization::domain::ids::lists, 83
is_domain_id_v
serif::discretization::domain::ids, 81
is_domain_id_v< DomainID< DomainT, IdentifierValue > >
serif::discretization::domain::ids, 81
is_domain_set_v
serif::discretization::domain, 79
is_domain_set_v< DomainSet< DomainTs... > >
serif::discretization::domain, 79
is_equation_of_state_model_v
serif::eos::models, 109
is_finite
serif::utils::checks, 194
is_geometry_cache_enabled
serif::mapping::DomainMapper, 473
is_key_list_v
serif::utils::dependencies, 203
is_key_list_v< KeyList< KeyTs... > >
serif::utils::dependencies, 203
is_linear
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 362
serif::geometry::deformation::RadialExtension, 825
serif::geometry::deformation::SurfaceEmbedding, 943
is_linear_deformation
serif::geometry::deformation, 126
is_owned_computation_v
serif::utils::dependencies, 203
is_owned_computation_v< OwnedComputation< KeyT, ComputationT > >
serif::utils::dependencies, 203
is_pressure_surface_formulation
serif::geometry::surface::detail, 147
is_pressure_surface_formulation< DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation > >
serif::geometry::surface::detail, 147
is_pressure_surface_formulation< PressureSurfaceFormulation< EOS, Bindings, Carrier > >
serif::geometry::surface::detail, 147
is_relation_list_v
serif::discretization::domain::relation::lists, 86
is_relation_list_v< RelationList< RelationTs... > >
serif::discretization::domain::relation::lists, 86
is_resource_v
serif::utils::dependencies, 203
is_resource_v< Resource< KeyT, StorageT, PolicyT > >
serif::utils::dependencies, 203
is_schema_v
serif::discretization::domain::schema, 89
is_schema_v< DomainSchema< DomainIDs, BoundaryIDs, Relations > >
serif::discretization::domain::schema, 89
is_thermodynamic_relation_v
serif::eos::relations, 112
is_valid_relation_catalog_v
serif::eos::relations, 112

Isobaric
serif::geometry::surface, 139
isobaric_surface
serif::discretization::quadrature::QuadratureOptions, 807
IsobaricSurface
serif::geometry::surface::IsobaricSurface, 647

J_element_to_physical_inverse
serif::mapping::VolumeQuadratureContext, 1022
jacobian
serif::mapping::DMFixedMeshGeometry, 431
JACOBIAN_ERROR
serif::utils::errors, 209
JACOBIAN_SHAPE_ERROR
serif::utils::errors, 209
JacobianEntries
serif::geometry::deformation::detail, 135
JacobianOperator
serif::operators::backend::mfem::JacobianOperator< CacheT >, 659
join
serif::utils::misc::format, 214

K
serif::eos::models::PolytropeParameters, 770
kelvin
serif::discretization::quadrature, 103
KelvinCompactification
serif::mapping::compactification::KelvinCompactification, 662
Key
serif::utils::dependencies::Binding< KeyT, TargetT >, 309
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >, 364
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >, 630
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >, 745
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >, 876
key_list_subset_v
serif::utils::dependencies::detail, 207
key_list_subset_v< KeyList< Keys... >, AvailableT >
serif::utils::dependencies::detail, 207
Keys
serif::utils::dependencies::KeyList< KeyTs >, 666
KinematicViscosityValue
serif::dimensions, 64

L2
serif::discretization::functional_space, 101
l2
serif::discretization::functional_space, 98
LengthValue
serif::dimensions, 64
limited_by_geometry
serif::geometry::deformation::GeometryStepLimit, 593
limiting_element
serif::geometry::deformation::GeometryStepLimit, 593
limiting_point
serif::geometry::deformation::GeometryStepLimit, 593
limiting_point_determinant_at_step
serif::geometry::deformation::GeometryStepLimit, 593
limiting_rule
serif::geometry::deformation::GeometryStepLimit, 593
linearize_carrier_samples
serif::geometry::surface::detail, 145
linearize_pressure_samples
serif::geometry::surface::detail, 145
linearize_surface
serif::geometry::surface::detail, 145
load
serif::mapping::VolumeMappingCache, 1016
load_element
serif::mapping::GridFunctionMappingEvaluator, 617
load_inverse_jacobian
serif::mapping::VolumeMappingCache, 1016
locate_boundary_point
serif::geometry::deformation::backend::mfem::detail, 133
logical_mesh
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza
831
logical_radii
serif::geometry::deformation::RadialInterpolation, 828
logical_radius
serif::geometry::deformation::backend::mfem::detail, 133
LuminosityValue
serif::dimensions, 64

m_active
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 371
m_angular_velocity
serif::equations::RigidRotation, 884
m_attributes

serif::discretization::backend::mfem::DofMap, 444
m_binding
serif::discretization::backend::mfem::DofSelection, 448
m_boundary_attributes
serif::discretization::backend::mfem::DofSelection, 448
m_boundaryElementBoundaryIDs
serif::discretization::domain::mesh::MeshTopology, 700
m_boundaryElementFaceIDs
serif::discretization::domain::mesh::MeshTopology, 700
m_boundaryElementsByFace
serif::discretization::domain::mesh::MeshTopology, 701
m_cache
serif::operators::backend::mfem::JacobianOperator< CacheT >, 660
m_cached_element_id
serif::mapping::GridFunctionMappingEvaluator, 618
m_center
serif::equations::RigidRotation, 884
m_coefficient
serif::equations::EquationScale< EquationT >, 521
m_compactification_coordinate
serif::mapping::GridFunctionMappingEvaluator, 618
m_compactification_data
serif::mapping::GridFunctionMappingEvaluator, 618
m_compactification_dofs
serif::mapping::GridFunctionMappingEvaluator, 618
m_compactification_dshape
serif::mapping::DMWorkspace, 435
m_compactification_point
serif::mapping::DMWorkspace, 435
m_compactification_shape
serif::mapping::DMWorkspace, 435
m_compactification_space
serif::mapping::GridFunctionMappingEvaluator, 618
m_compactification_space_sequence
serif::mapping::GridFunctionMappingEvaluator, 618
m_components
serif::discretization::backend::mfem::BasisEvaluation, 298
serif::discretization::backend::mfem::TensorBasis, 954
serif::operators::backend::mfem::BasisEvaluation, 306
m_computations
serif::utils::dependencies::ComputationGraph< Computations >, 378
m_correction
serif::utils::CompensatedSum, 357
m_data
serif::mapping::VolumeMappingCache, 1016
m_dense
serif::discretization::backend::mfem::BasisEvaluation, 298
serif::operators::backend::mfem::BasisEvaluation, 306
m_diagnosticControl
serif::discretization::quadrature::RuleSet, 894
m_dimension
serif::mapping::DMWorkspace, 435
serif::mapping::ElementDisplacementData, 489
serif::mapping::VolumeMappingCache, 1016
m_discretizationControl
serif::discretization::quadrature::RuleSet, 894
m_displacement
serif::mapping::GridFunctionMappingEvaluator, 618
m_displacement_data
serif::mapping::GridFunctionMappingEvaluator, 619
m_displacement_dofs
serif::mapping::GridFunctionMappingEvaluator, 619
m_displacement_space
serif::mapping::GridFunctionMappingEvaluator, 619
m_displacement_space_sequence
serif::mapping::GridFunctionMappingEvaluator, 619
m_dof_matrix
serif::mapping::ElementDisplacementData, 489
m_dofs
serif::discretization::backend::mfem::DofSelection, 448
serif::discretization::backend::mfem::TensorBasis, 954
serif::mapping::ElementCompactificationData, 486
m_domains
serif::discretization::domain::DomainOrSet, 476
m_element
serif::mapping::ElementCompactificationData, 486
serif::mapping::ElementDisplacementData, 489
m_element_compactification
serif::mapping::GridFunctionMappingEvaluator,

619
m_element_displacement
serif::mapping::GridFunctionMappingEvaluator, 619
m_element_support
serif::discretization::backend::mfem::DofMap, 444
m_elementDomainIDs
serif::discretization::domain::mesh::MeshTopology, 701
m_elementNeighbors
serif::discretization::domain::mesh::MeshTopology, 701
m_enthalpy_power
serif::equations::BarotropicClosure< EOST >, 284
serif::equations::PressureForce< EOST >, 789
m_enthalpy_scale
serif::eos::models::Polytrope, 769
m_entries
serif::discretization::backend::mfem::TensorBasisCache, 957
serif::operators::backend::mfem::TensorBasisCache, 959
serif::utils::dependencies::ComputationStore< EntryTs >, 382
m_eos
serif::equations::BarotropicClosure< EOST >, 284
serif::equations::CentralDensityConstraint< EOST >, 342
serif::equations::PressureForce< EOST >, 789
m_equation
serif::equations::DisplacementForce< EOST >, 427
serif::equations::EquationDifference< LeftT, RightT >, 511
serif::equations::EquationScale< EquationT >, 521
m_equations
serif::equations::EquationRegistry< EquationTs >, 518
serif::equations::EquationSum< FirstT, RestTs >, 524
serif::operators::EquationRegistry< EquationTs >, 519
m_evaluate
serif::eos::runtime::EOSView, 509
m_extension
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 362
m_exterior_map
serif::mapping::DomainMapper, 474
m_exterior_result
serif::mapping::DMWorkspace, 435
m_exterior_variation
serif::mapping::DMWorkspace, 436
m_f
serif::mapping::PhysicalPositionFunctionCoefficient, 757
m_faceElements
serif::discretization::domain::mesh::MeshTopology, 701
m_fallbackControl
serif::discretization::quadrature::RuleSet, 894
m_field_jacobian
serif::mapping::DMWorkspace, 436
m_field_value
serif::mapping::DMWorkspace, 436
m_formRules
serif::discretization::quadrature::RuleSet, 894
m_four_pi_G
serif::equations::PoissonBalance, 764
m_full_element_jacobian
serif::mapping::DMWorkspace, 436
m_full_to_reduced
serif::discretization::backend::mfem::DofMap, 444
m_function
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >, 364
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >, 631
serif::utils::dependencies::ProjectedSnapshot< FunctionT >, 801
m_geometryCache
serif::mapping::DomainMapper, 474
m_graph
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 371
m_hessian
serif::equations::RigidRotation, 884
m_interpolation
serif::geometry::deformation::RadialExtension, 825
m_mapped_normal
serif::mapping::DMWorkspace, 436
m_mapper
serif::mapping::GridFunctionMappingEvaluator, 619
m_mapping
serif::discretization::backend::mfem::BasisEvaluation, 299
serif::mapping::PhysicalPositionFunctionCoefficient, 757
serif::operators::backend::mfem::BasisEvaluation, 306
m_matrix_temp_1

serif::mapping::DMWorkspace, 436
m_matrix_temp_2
serif::mapping::DMWorkspace, 436
m_mesh
serif::discretization::backend::mfem::DofMap, 444
m_mesh_dshape
serif::mapping::DMWorkspace, 436
m_mesh_sequence
serif::discretization::backend::mfem::DofMap, 444
m_name
serif::dimensions::runtime::ThermodynamicQuantityID, 975
serif::mapping::compactification::KelvinCompactification, 664
m_nodes_sequence
serif::discretization::backend::mfem::DofSelection, 449
m_object
serif::eos::runtime::EOSView, 509
m_options
serif::mapping::compactification::KelvinCompactification, 664
serif::mapping::DomainMapper, 474
m_ordering
serif::mapping::ElementDisplacementData, 490
m_parent
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >, 417
m_partialDerivative
serif::eos::runtime::EOSView, 509
m_point_count
serif::mapping::VolumeMappingCache, 1017
m_point_stride
serif::mapping::VolumeMappingCache, 1017
m_points
serif::discretization::backend::mfem::BasisEvaluation, 299
serif::discretization::backend::mfem::TensorBasis, 955
serif::operators::backend::mfem::BasisEvaluation, 306
m_policy
serif::discretization::quadrature::backend::mfem::MFEMResolver, 707
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >, 877
m_polytropic_constant
serif::eos::models::Polytrope, 769
m_polytropic_index
serif::eos::models::Polytrope, 769
m_preconditionerControl
serif::discretization::quadrature::RuleSet, 894
m_preparation_count
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 371
m_products
serif::discretization::backend::mfem::TensorBasis, 955
m_projectionControl
serif::discretization::quadrature::RuleSet, 894
m_radial_weights
serif::geometry::deformation::RadialExtension, 825
m_ready
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 371
m_reduced_to_full
serif::discretization::backend::mfem::DofMap, 444
m_reference_dshape
serif::mapping::DMWorkspace, 436
m_reference_field_jacobian
serif::mapping::DMWorkspace, 436
m_reference_normal
serif::mapping::DMWorkspace, 437
m_relations
serif::eos::runtime::EOSView, 509
m_report
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 372
m_resources
serif::utils::dependencies::DependencyContext< ResourceTs >, 415
m_rotation
serif::equations::AngularMomentum, 268
serif::equations::BarotropicHydrostaticEquilibrium, 287
serif::equations::RotationalForce, 891
serif::operators::BarotropicHydrostaticEquilibrium, 289
m_ruleSet
serif::discretization::quadrature::Policy, 766
m_shape
serif::mapping::DMWorkspace, 437
m_space
serif::discretization::backend::mfem::DofMap, 444
m_space_sequence
serif::discretization::backend::mfem::DofMap, 445
m_storage
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 372
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >, 877
m_sum

serif::utils::CompensatedSum, 357
m_surface
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 362
m_target_density
serif::equations::CentralDensityConstraint< EOST >, 342
m_target_enthalpy
serif::equations::CentralDensityConstraint< EOST >, 342
serif::equations::PressureSurfaceConstraint< EOST >, 792
m_target_mass
serif::equations::MassNormalization, 695
m_target_momentum
serif::equations::AngularMomentum, 268
m_target_pressure
serif::equations::PressureSurfaceConstraint< EOST >, 792
m_targetPressure
serif::geometry::surface::IsobaricSurface, 647
m_tensor
serif::discretization::backend::mfem::BasisEvaluation, 299
serif::operators::backend::mfem::BasisEvaluation, 306
m_value
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >, 755
serif::dimensions::QuantityValue< Quantity >, 820
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >, 747
m_vector_temp
serif::mapping::DMWorkspace, 437
m_width
serif::discretization::backend::mfem::BasisEvaluation, 299
serif::operators::backend::mfem::BasisEvaluation, 306
m_workspace
serif::mapping::GridFunctionMappingEvaluator, 619
serif::operators::backend::mfem::JacobianOperator< CacheT >, 660
MAGENTA
serif::utils::misc::terminal, 217
MagneticFluxDensityValue
serif::dimensions, 64
make
serif::discretization::backend::mfem::TensorBasis, 953
make_attribute_marker
serif::discretization::domain::schema, 89
make_boundary_discretization
serif::operators::backend::mfem, 180
make_computation
serif::utils::dependencies, 199
make_computation_cache
serif::utils::dependencies, 199
make_computation_graph
serif::utils::dependencies, 200
make_computation_store
serif::utils::dependencies, 200
make_contribution_computations
serif::operators::backend::mfem, 180
make_deformation_computations
serif::geometry::deformation, 125
serif::geometry::deformation::backend::mfem, 129
make_deformation_graph
serif::geometry::deformation, 125
serif::geometry::deformation::backend::mfem, 129
make_deformation_inputs
serif::geometry::deformation::backend::mfem, 130
make_deformation_workspace
serif::geometry::deformation, 125
make_dependency_context
serif::utils::dependencies, 200
make_harmonic_modes
serif::geometry::deformation::detail, 136
make_incremental_computation
serif::utils::dependencies, 200
make_MappingStatus_SERiFErrorCode_map
serif::mapping, 151
make_nodal_dependency_context
serif::operators::backend::mfem, 180
make_nodal_discretization
serif::operators::backend::mfem, 180
make_nodal_equation_cache
serif::operators::backend::mfem, 181
make_nodal_equation_computation
serif::operators::backend::mfem, 181
make_nodal_equation_point
serif::operators::backend::mfem, 181
make_operand_computations
serif::operators::backend::mfem, 181
make_order
serif::utils::dependencies::detail::Graph< Computations >, 606
make_point_border_cache
serif::operators::backend::mfem, 181
make_point_border_dependency_context
serif::operators::backend::mfem, 182
make_point_border_discretization
serif::operators::backend::mfem, 182

make_point_discretization
serif::operators::backend::mfem, 182
make_pressure_surface_graph
serif::geometry::surface, 140
make_query
serif::discretization::quadrature, 104
make_radial_extension
serif::geometry::deformation, 125
serif::geometry::deformation::RadialExtension, 825
make_SERIFErrorCode_MappingStatus_map
serif::mapping, 151
make_surface_eos_computation
serif::geometry::surface::detail, 146
make_surface_linearization_computation
serif::geometry::surface::detail, 146
make_surface_metric_computation
serif::geometry::deformation::backend::mfem, 130
make_term_coefficients
serif::operators::backend::mfem, 182
make_volume_displacement_computations
serif::geometry::deformation::backend::mfem, 130, 131
make_volume_equation_cache
serif::operators::backend::mfem, 182
make_volume_equation_computations
serif::operators::backend::mfem, 183
make_volume_inputs
serif::operators::backend::mfem, 183, 184
make_workspace
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 361
makePartialDerivativeMask
serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >, 905
manifest
serif::discretization::quadrature::QuadratureOptions, 808
map
serif::geometry::deformation::DeformationLinearization< Map >, 404
map_basis_point
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 731
map_contribution_geometry
serif::operators::backend::mfem, 184
map_contribution_operand
serif::operators::backend::mfem, 184
map_element_geometry
serif::operators::backend::mfem, 185
map_element_operand
serif::operators::backend::mfem, 185
map_hcurl_curl_to_physical
serif::mapping, 151
map_hcurl_field_to_physical
serif::mapping, 151
map_hdiv_divergence_to_physical
serif::mapping, 151
map_hdiv_flux_to_physical
serif::mapping, 151
map_physical_curl_to_hcurl_reference
serif::mapping, 152
map_physical_field_to_hcurl_reference
serif::mapping, 152
map_physical_flux_to_hdiv_reference
serif::mapping, 152
map_physical_gradient_to_reference
serif::mapping, 152
map_physical_vector_gradient_to_reference
serif::mapping, 152
map_point
serif::discretization::backend::mfem::BasisEvaluation, 296
serif::operators::backend::mfem::BasisEvaluation, 304
map_reference_gradient_to_physical
serif::mapping, 152
map_reference_vector_gradient_to_physical
serif::mapping, 153
mapper
serif::operators::backend::mfem::BasisBindings< ColumnCount >, 291
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >, 1013
MAPPING
serif::operators::backend::mfem, 173
Mapping
serif::discretization::functional_space, 98
mapping
serif::discretization::functional_space::SpaceDescriptor, 919
serif::discretization::quadrature::backend::mfem::MFEMQuadratu
706
serif::discretization::quadrature::Query, 821
serif::mapping::FaceMappingContext, 530
serif::mapping::FaceMappingVariation, 531
serif::mapping::VolumeMappingContext, 1018
serif::mapping::VolumeMappingVariation, 1019
serif::operators::backend::mfem::ElementGeometry< OperandCount >, 491
Mapping2StringMap
serif::discretization::functional_space, 101
mapping_context
serif::operators::backend::mfem::Workspace, 1028
mapping_determinate

serif::mapping::MappingPointContext, 677
serif::mapping::VolumeQuadratureContext, 1022
mapping_determinate_variation
serif::mapping::MappingPointVariation, 678
mapping_jacobian
serif::mapping::compactification::ExteriorMapResult, 528
serif::mapping::MappingPointContext, 677
mapping_jacobian_variation
serif::mapping::compactification::ExteriorMapVariation, 528
serif::mapping::MappingPointVariation, 678
mapping_status
serif::operators::backend::mfem::PreparationRejection, 781
mapping_variation
serif::operators::backend::mfem::Workspace, 1028
mapping_workspace
serif::operators::backend::mfem::Workspace, 1028
MappingKind
serif::discretization::quadrature, 103
MappingStatus
serif::mapping, 150
MappingStatus_enum_string_map
serif::mapping, 153
MappingStatus_SERiFErrorCode_map
serif::mapping, 153
MappingStatus_string_enum_map
serif::mapping, 153
Mask
serif::utils::dependencies::detail::Graph< Computations >, 606
mask_residual_dual
serif::operators::backend::mfem, 185
Mass
serif::operators, 158
mass_conservation
serif::discretization::quadrature::QuadratureOptions, 808
mass_density_term
serif::discretization::blocks::density, 410
mass_normalization
serif::discretization::quadrature::QuadratureOptions, 808
mass_normalization_term
serif::discretization::blocks::barotropic_constant, 282
MassFlowRateValue
serif::dimensions, 64
MassNormalization
serif::equations::MassNormalization, 694
MassValue
serif::dimensions, 64
Matches
serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Relations... > >, 696
matches
serif::discretization::backend::mfem::CoefficientTracking, 349
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza, 834
serif::geometry::surface::detail::PressureConditionTracking, 786
serif::operators::backend::mfem::CoefficientTracking, 352
serif::utils::dependencies::ProjectedSnapshot< FunctionT >, 801
serif::utils::dependencies::ValueSnapshot, 999
matches_pressure_relation
serif::geometry::surface::detail, 146
Matrix
serif::equations::RigidRotation, 883
serif::utils::dependencies::detail::Graph< Computations >, 606
MATRIX_SHAPE_ERROR
serif::utils::errors, 210
MatrixValue
serif::discretization::forms, 94
maximum_azimuthal_order
serif::geometry::deformation::SphericalHarmonicSurface, 934
maximum_degree
serif::geometry::deformation::SphericalHarmonicSurface, 934
maximum_step
serif::geometry::deformation::GeometryStepOptions, 594
Mechanical
serif::operators, 158
MechanicalDifference
serif::operators, 158
mesh
serif::discretization::backend::mfem::DofMap, 441
serif::mapping::DMGeometryCache, 432
mesh_extension
serif::discretization::quadrature::QuadratureOptions, 808
meshes
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza, 833
MeshTopology
serif::discretization::domain::mesh::MeshTopology, 699
message

serif::geometry::deformation::DeformationFailure, 403
serif::geometry::surface::SurfaceFailure, 946
serif::operators::backend::mfem::PreparationRejection, 781
mfem, 57
mfem_form_static_order
serif::discretization::quadrature::backend::mfem, 106
MFEMResolver
serif::discretization::quadrature::backend::mfem::MFEMResolver, 707
minimum_determinant_at_accepted
serif::geometry::deformation::GeometryStepLimit, 593
minimum_determinant_at_maximum
serif::geometry::deformation::GeometryStepLimit, 593
MISC_MAPPING_ERROR
serif::utils::errors, 209
MISMATCHED_MESHING_MAPS
serif::utils::errors, 209
misses
serif::mapping::DMGeometryCache, 432
serif::mapping::DMGeometryCacheStatistics, 433
MISSING_DISPLACEMENT_FINITE_ELEMENT_SPACE
serif::utils::errors, 209
MISSING_EXTERIOR_MAP
serif::utils::errors, 209
MISSING_FACE_ELEMENT
serif::utils::errors, 209
MixedSamples
serif::operators, 158
Mode
serif::discretization::quadrature, 103
mode
serif::discretization::quadrature::QuadratureOptions, 808
modes
serif::geometry::deformation::SurfaceEmbedding, 942
modes_
serif::geometry::deformation::SurfaceEmbedding, 943
MomentOfInertiaValue
serif::dimensions, 64
Momentum
serif::operators, 158
MomentumValue
serif::dimensions, 65
Mult
serif::operators::backend::mfem::JacobianOperator< CacheT >, 659
MultTranspose
serif::operators::backend::mfem::JacobianOperator< CacheT >, 660

n
serif::eos::models::PolytropeParameters, 770
name
serif::dimensions::runtime::ThermodynamicQuantityID, 975
serif::discretization::blocks::barotropic_constant, 282
serif::discretization::blocks::barotropic_constant::mass_normaliz 689
serif::discretization::blocks::central_density, 339
serif::discretization::blocks::central_density::border, 313
serif::discretization::blocks::density, 410
serif::discretization::blocks::density::mass, 683
serif::discretization::blocks::displacement, 425
serif::discretization::blocks::displacement::vector, 1003
serif::discretization::blocks::enthalpy, 497
serif::discretization::blocks::enthalpy::specific, 925
serif::discretization::blocks::FieldDescriptor, 535
serif::discretization::blocks::gravity, 611
serif::discretization::blocks::gravity::gradient, 598
serif::discretization::blocks::gravity::potential, 773
serif::discretization::blocks::rotation, 887
serif::discretization::blocks::rotation::angular_velocity, 262
serif::discretization::blocks::surface_deformation, 938
serif::discretization::blocks::surface_deformation::coefficients, 347
serif::discretization::blocks::surface_deformation::shape_equilibr 914
serif::discretization::blocks::TermDescriptor, 969
serif::discretization::domain::CoreDomain, 398
serif::discretization::domain::EnvelopeDomain, 504
serif::discretization::domain::ids::BoundaryDescriptor, 319
serif::discretization::domain::ids::DomainDescriptor, 459
serif::discretization::domain::InfinitySurfaceBoundary, 632
serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >, 451
serif::discretization::domain::relation::FullyConnected< A >, 585

serif::discretization::domain::relation::Inscribed< A, B >, 635
serif::discretization::domain::schema::validation::DomainBoundary< BoundaryT, DomainTs >, 453
serif::discretization::domain::schema::validation::FullyConnected< A >, 587
serif::discretization::domain::schema::validation::Inscribed< A, B >, 637
serif::discretization::domain::StellarSurfaceBoundary, 936
serif::discretization::domain::VacuumDomain, 980
serif::discretization::forms::boundary_flux< TrialT, TestT >, 318
serif::discretization::forms::divergence< TrialT, TestT >, 430
serif::discretization::forms::force< TrialT, TestT >, 541
serif::discretization::forms::FormDescriptor, 572
serif::discretization::forms::mass< TrialT, TestT >, 687
serif::discretization::forms::projection< TrialT, TestT >, 804
serif::discretization::forms::source< TrialT, TestT >, 917
serif::equations::AngularMomentum, 268
serif::equations::BarotropicClosure< EOST >, 284
serif::equations::BarotropicHydrostaticEquilibrium, 287
serif::equations::CenteringConstraint, 337
serif::equations::CentralDensityConstraint< EOST >, 342
serif::equations::DisplacementForce< EOST >, 427
serif::equations::EquationDifference< LeftT, RightT >, 511
serif::equations::EquationScale< EquationT >, 522
serif::equations::EquationSum< FirstT, RestTs >, 524
serif::equations::GravityForce, 613
serif::equations::GravityGradientRelation, 615
serif::equations::MassNormalization, 695
serif::equations::PoissonBalance, 764
serif::equations::PressureForce< EOST >, 790
serif::equations::PressureSurfaceConstraint< EOST >, 792
serif::equations::RotationalForce, 891
serif::operators::BadContributionCollection, 278
serif::operators::BadContributionRow, 280
serif::operators::BarotropicHydrostaticEquilibrium, 289
serif::operators::GradientRequest, 602
serif::operators::NestedBadCollection, 713
serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >, 719
names
serif::discretization::domain::DomainSet< DomainTs >, 481
NEGATIVE_VALUE_ERROR
serif::utils::errors, 209
NeitherConstNorVolatile
serif::dimensions, 65
serif::utils::misc::std_helper, 214
NeitherConstNorVolatileReference
serif::dimensions, 65
serif::utils::misc::std_helper, 214
nodal
serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >, 760
serif::operators::backend::mfem::PointBorderDiscretization< EquationT, SchemaT >, 761
nodal_basis
serif::geometry::deformation::SurfaceEmbedding, 942
nodal_basis_
serif::geometry::deformation::SurfaceEmbedding, 943
nodal_work_report
serif::operators::backend::mfem, 185
NodalCache
serif::operators::backend::mfem, 170
NodalCoefficients
serif::operators::backend::mfem, 170
NodalMesh
serif::operators::backend::mfem, 170
nodes_sequence
serif::geometry::deformation::backend::mfem::ReferenceMeshSnap 836
serif::operators::backend::mfem::BasisBindings< ColumnCount >, 291
nodesSequence
serif::mapping::DMGeometryCache, 432
NON_FINITE
serif::operators::backend::mfem, 173
non_finite
serif::geometry::deformation, 122
NON_FINITE_DETERMINATE
serif::utils::errors, 209
non_finite_eos_response
serif::geometry::surface, 139
non_finite_linearization
serif::geometry::surface, 139
NON_FINITE_MAPPING_INPUT
serif::mapping, 150
serif::utils::errors, 209
NON_FINITE_MAPPING_RESULT

serif::mapping, 150
serif::utils::errors, 209
non_finite_state
serif::geometry::surface, 139
NON_FINITE_VALUE_ERROR
serif::utils::errors, 209
NON_POSITIVE_DETERMINATE
serif::mapping, 150
serif::utils::errors, 209
None
serif::discretization::domain::relation::validation, 87
none
serif::discretization::functional_space, 99
serif::discretization::quadrature, 103
NonPressureInputs
serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > >, 845
normal
serif::discretization::functional_space, 99
serif::mapping::FaceQuadratureContext, 532
normal_continuous
serif::discretization::functional_space, 98
normal_flux_scale_variation
serif::mapping::FaceMappingVariation, 531
not_required
serif::utils::dependencies, 199
NumberDensityValue
serif::dimensions, 65

Omega
serif::operators, 158
OpacityValue
serif::dimensions, 65
Operand
serif::equations::gravity_contributions, 118
operand_basis
serif::operators::backend::mfem::ElementGeometry< OperandCount >, 491
operand_count
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 515
operand_dual
serif::operators::backend::mfem::Workspace, 1028
operand_types
serif::discretization::forms::boundary_flux< TrialT, TestT >, 317
serif::discretization::forms::divergence< TrialT, TestT >, 430
serif::discretization::forms::force< TrialT, TestT >, 541
serif::discretization::forms::mass< TrialT, TestT >, 687
serif::discretization::forms::projection< TrialT, TestT >, 803
serif::discretization::forms::source< TrialT, TestT >, 916
operand_values
serif::operators::backend::mfem::Workspace, 1028
Operands
serif::equations::AngularMomentum, 267
serif::equations::BarotropicClosure< EOST >, 283
serif::equations::BarotropicHydrostaticEquilibrium, 286
serif::equations::CenteringConstraint, 336
serif::equations::CentralDensityConstraint< EOST >, 341
serif::equations::gravity_contributions::DensitySource, 412
serif::equations::gravity_contributions::GradientDivergence, 600
serif::equations::gravity_contributions::GradientValue, 604
serif::equations::gravity_contributions::PotentialValue, 775
serif::equations::GravityForce, 613
serif::equations::MassNormalization, 694
serif::equations::PressureForce< EOST >, 788
serif::equations::PressureSurfaceConstraint< EOST >, 791
serif::equations::RotationalForce, 890
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 794
serif::operators::BarotropicHydrostaticEquilibrium, 288
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 513
serif::operators::GradientRequest, 601
serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >, 719
OperandValue
serif::discretization::forms, 94
operation
serif::discretization::backend::mfem::TensorBasisCache::Entry, 500
serif::operators::backend::mfem::TensorBasisCache::Entry, 501
operation_type
serif::discretization::forms::Operand< TermT, OperationT >, 727
operator bool
serif::discretization::domain::relation::validation::RelationValidati

850
serif::discretization::domain::schema::validation::RelationValidationResult, 851
serif::discretization::domain::schema::validation::SchemaRelationValidationResult, 907
serif::discretization::domain::schema::validation::SchemaValidationResult, 909
operator int
serif::discretization::blocks::residual_block< index_value >, 867
serif::discretization::blocks::value_block< index_value >, 996
operator<=>
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >, 754, 755
serif::dimensions::QuantityValue< Quantity >, 818, 819
operator+
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >, 752
serif::dimensions::QuantityValue< Quantity >, 816
operator-
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >, 753
serif::dimensions::QuantityValue< Quantity >, 816
operator/
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >, 753
serif::dimensions::QuantityValue< Quantity >, 816
operator=
serif::discretization::backend::mfem::DofMap, 442
serif::mapping::DomainMapper, 473
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 369
serif::utils::dependencies::ComputationStore< EntryTs >, 382
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >, 747
operator==
serif::dimensions::QuantityValue< Quantity >, 819
serif::dimensions::runtime::ThermodynamicQuantityDT, 975
serif::discretization::blocks::FieldDescriptor, 535
serif::discretization::blocks::TermDescriptor, 969
serif::discretization::forms::FormDescriptor, 572
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot, 833
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot, 836
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot, 837
serif::geometry::deformation::DomainDeformation< SurfaceDeformation, VacuumT >, 457
serif::geometry::deformation::FixedInfinityRadialExtension, 737
serif::geometry::deformation::NodalRadialSurface, 717
serif::geometry::deformation::PowerLawRadialExtension, 777
serif::geometry::deformation::SphericalHarmonicMode, 933
serif::geometry::deformation::SphericalHarmonicSurface, 934
serif::geometry::surface::BoundaryValues< Term >, 328
serif::operators::backend::mfem::BasisBindings< ColumnCount >, 291
operator*
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >, 752
serif::dimensions::QuantityValue< Quantity >, 815
options
serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >, 1008
Order
serif::utils::dependencies::detail::Graph< Computations >, 606
order
serif::discretization::blocks::SpatialTermTraits< TermT >, 922
serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >, 422
serif::discretization::quadrature::backend::mfem::DiscreteFunctional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > >, 423
serif::discretization::quadrature::Resolution, 870
serif::geometry::deformation::SphericalHarmonicMode, 933
serif::utils::dependencies::detail::Graph< Computations >, 607
ordering
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot, 837
OUT_OF_RANGE_ERROR
serif::utils::errors, 210
outer_type
serif::discretization::domain::relation::Inscribed< A, B >, 634
serif::discretization::domain::schema::validation::Inscribed< A, B >, 636
OuterDomainAbstract
serif::discretization::domain::relation::validation,

87
output
serif::discretization::backend::mfem::TensorBasis::Producer, 800
output_file
serif::discretization::quadrature::QuadratureManifestOptions, 805
output_size
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 361
serif::geometry::deformation::RadialExtension, 824
serif::geometry::deformation::SurfaceEmbedding, 942
OutputQuantity
serif::eos::relations::Relation< Output, Inputs >, 840
outputQuantity
serif::dimensions::runtime::RuntimeRelationDescriptor, 901
OutputSpace
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 360
serif::geometry::deformation::RadialExtension, 823
serif::geometry::deformation::SurfaceEmbedding, 941
OUTSIDE_REFERENCE_DOMAIN
serif::mapping, 150
serif::utils::errors, 209
own
serif::utils::dependencies, 200
own_computation
serif::utils::dependencies, 201
OwnedComputation
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >, 746

padding
serif::utils::types::ct_error_string< N >, 400
parameter_pullback
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace, 1025
parameters
serif::geometry::deformation::DeformationLinearization< Map >, 404
parent
serif::operators::detail::ContributionChildren< EquationT >, 390
parse
std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult >, 547
partial_derivative
serif::eos, 107
serif::eos::models::Polytrope, 768, 769
serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > >, 895
PartialDerivative
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >, 751
serif::discretization::forms, 94
partialDerivativeMask
serif::dimensions::runtime::RuntimeRelationDescriptor, 901
serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >, 906
passed_relation_count
serif::discretization::domain::schema::validation::SchemaValidatio 909
physical
serif::operators::backend::mfem::OperandSamples, 736
physical_integration_weight
serif::mapping::VolumeQuadratureContext, 1022
physical_normal_variation
serif::mapping::FaceMappingVariation, 531
physical_position
serif::mapping::compactification::ExteriorMapResult, 528
serif::mapping::MappingPointContext, 677
physical_position_variation
serif::mapping::compactification::ExteriorMapVariation, 528
serif::mapping::MappingPointVariation, 679
physical_surface_weight
serif::mapping::FaceMappingContext, 530
physical_surface_weight_variation
serif::mapping::FaceMappingVariation, 531
physical_value
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 731
PhysicalCondition
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 795
PhysicalPositionFunctionCoefficient
serif::mapping::PhysicalPositionFunctionCoefficient, 757
PhysicalQuantity
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 795
serif::geometry::surface::IsobaricSurface, 646
physics_points
serif::operators::backend::mfem::NumericalWork,

724
Plan
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >, 1011
point
serif::geometry::deformation::backend::mfem::detail::Locate, 673
serif::geometry::deformation::DeformationFailure, 403
points
serif::discretization::backend::mfem::TensorBasis, 954
serif::discretization::backend::mfem::TensorBasisCache::entry, 500
serif::mapping::DMGeometryCache, 432
serif::mapping::DMGeometryCacheStatistics, 433
serif::operators::backend::mfem::TensorBasisCache::entry, 502
PoissonBalance
serif::equations::PoissonBalance, 764
Policy
serif::discretization::quadrature::Policy, 765
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >, 876
policy
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >, 877
Polytrope
serif::eos::models::Polytrope, 767
polytropic_constant
serif::eos::models::Polytrope, 769
polytropic_index
serif::eos::models::Polytrope, 769
position
serif::equations::EquationPoint< OperandsT >, 516
serif::mapping::DMFixedMeshGeometry, 431
position_derivative
serif::equations::PointLinearization< OperandsT, ResultT >, 762
position_independent
serif::equations::CenteringConstraint, 337
serif::equations::CentralDensityConstraint< EOST >, 342
serif::equations::PressureSurfaceConstraint< EOST >, 793
serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >, 719
positions
serif::geometry::deformation::ReferenceSurface, 839
Potential
serif::equations::BarotropicHydrostaticEquilibrium, 286
serif::equations::gravity_contributions, 118
serif::operators, 158
serif::operators::BarotropicHydrostaticEquilibrium, 288
potential
serif::equations::HorizontalBoundaryPoint, 884
serif::equations::RigidRotation, 884
potential_term
serif::discretization::blocks::gravity, 611
PowerValue
serif::dimensions, 65
preconditioner
serif::discretization::quadrature, 104
serif::discretization::quadrature::QuadratureRoleOptions, 810
serif::discretization::quadrature::RoleControls, 886
preparation_count
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 370
preparation_mode_name_map
serif::utils::dependencies, 203
preparation_status_name_map
serif::utils::dependencies, 203
PreparationMode
serif::operators::backend::mfem, 171
serif::utils::dependencies, 198
PreparationRejectionReason
serif::operators::backend::mfem, 172
PreparationStatus
serif::utils::dependencies, 198
prepare_in_order
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 370
prepare_nodal
serif::operators::backend::mfem, 185
prepare_one
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 370
prepare_point_border
serif::operators::backend::mfem, 186
prepare_volume
serif::operators::backend::mfem, 186
PrependT
serif::utils::misc::concepts, 213
prerequisites
serif::utils::dependencies::detail::Graph< Computations >, 606
Pressure
serif::operators, 158
pressure_condition
serif::geometry::surface, 141
pressure_force

serif::discretization::quadrature::QuadratureOptions, 808
pressure_integral
serif::discretization::quadrature::QuadratureOptions, 808
pressure_relation_t
serif::geometry::surface, 139
PressureForce
serif::equations::PressureForce< EOST >, 789
PressureFromDensity
serif::eos::relations, 111
PressureFromSpecificEnthalpy
serif::eos::relations, 111
PressureSurfaceConstraint
serif::equations::PressureSurfaceConstraint< EOST >, 792
PressureValue
serif::dimensions, 65
previous
serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >, 383
production
serif::discretization::quadrature, 104
ProjectedSnapshot
serif::utils::dependencies::ProjectedSnapshot< FunctionT >, 800
projection
serif::discretization::quadrature, 104
serif::discretization::quadrature::QuadratureRoleOptions, 810
serif::discretization::quadrature::RoleControls, 886
PSurf
serif::geometry::surface::IsobaricSurface::Parameters, 749

quadrature
serif::mapping::FaceMappingContext, 530
serif::mapping::VolumeMappingContext, 1018
quadrature_point
serif::operators::backend::mfem::PreparationRejection, 781
quadrature_policy
serif::operators::backend::mfem::VolumeOptions, 1022
quadrature_values
serif::operators::backend::mfem::Workspace, 1028
QuadratureRole
serif::discretization::quadrature, 104
quadrupole
serif::discretization::quadrature::QuadratureOptions, 808
Quantity
serif::discretization::blocks::barotropic_constant::mass_normaliz 689
serif::discretization::blocks::central_density::border, 313
serif::discretization::blocks::density::mass, 683
serif::discretization::blocks::displacement::vector, 1002
serif::discretization::blocks::enthalpy::specific, 925
serif::discretization::blocks::gravity::gradient, 598
serif::discretization::blocks::gravity::potential, 772
serif::discretization::blocks::rotation::angular_velocity, 262
QuantityOf
serif::discretization::blocks, 75
QuantityOfT
serif::dimensions, 66
QuantityValue
serif::dimensions::QuantityValue< Quantity >, 814

r_inf_ref
serif::utils::user_options::KelvinCompactificationOptions, 665
r_star_ref
serif::utils::user_options::KelvinCompactificationOptions, 665
radial_directions
serif::geometry::deformation::SurfaceEmbedding, 942
radial_directions_
serif::geometry::deformation::SurfaceEmbedding, 943
radial_power
serif::geometry::deformation::PowerLawRadialExtension, 778
radial_weights
serif::geometry::deformation::RadialExtension, 825
RadialExtension
serif::geometry::deformation::RadialExtension, 824
rank
serif::discretization::blocks::SpatialTermTraits< TermT >, 922
serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >, 422
serif::discretization::quadrature::backend::mfem::DiscreteFunctio functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > >, 423
serif::utils::types::DenseTensorBase< T, Rank >, 408

serif::utils::types::SparseTensorBase< T, Rank >, 921
ready
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 370
reason
serif::geometry::deformation::DeformationFailure, 403
serif::geometry::surface::SurfaceFailure, 946
serif::operators::backend::mfem::PreparationRejection, 781
rebuild
serif::utils::dependencies, 198
serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >, 384
RED
serif::utils::misc::terminal, 217
reduced_dof
serif::discretization::backend::mfem::DofMap, 442
reduced_dofs
serif::discretization::backend::mfem::DofSelection, 447
reference
serif::discretization::quadrature, 104
serif::operators::backend::mfem::OperandSamples, 736
reference_center
serif::geometry::deformation::NodalRadialSurface, 717
serif::geometry::deformation::SphericalHarmonicSurface, 934
reference_components
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 733
reference_normal
serif::mapping::FaceMappingContext, 530
reference_operand_basis
serif::operators::backend::mfem::ElementBasis< OperandCount >, 484
reference_operand_values
serif::operators::backend::mfem::ElementData< OperandCount >, 487
reference_position
serif::equations::CenteringConstraint, 337
serif::equations::CentralDensityConstraint< EOST >, 343
serif::mapping::compactification::ExteriorMapInput, 527
serif::mapping::MappingPointContext, 677
reference_positions
serif::geometry::deformation::SurfaceEmbedding, 942
reference_positions_
serif::geometry::deformation::SurfaceEmbedding, 943
reference_radii
serif::geometry::deformation::SurfaceEmbedding, 942
reference_radii_
serif::geometry::deformation::SurfaceEmbedding, 944
reference_surface_weight
serif::mapping::FaceMappingContext, 530
reference_test_basis
serif::operators::backend::mfem::ElementBasis< OperandCount >, 484
reference_test_values
serif::operators::backend::mfem::Workspace, 1028
reference_value
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 731
reference_values
serif::discretization::backend::mfem::BasisWorkspace, 307
serif::operators::backend::mfem::BasisWorkspace, 308
ReferenceOperation
serif::discretization::backend::mfem, 70
serif::operators::backend::mfem, 173
ReferenceValue
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 729
refresh
serif::mapping::GridFunctionMappingEvaluator, 618
refreshed_scalars
serif::operators::backend::mfem::NumericalWork, 724
registered
serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > >, 325
serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > >, 326
serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... > >, 466
serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > >, 467
reject_negative_boosts
serif::discretization::quadrature::QuadratureValidationOptions, 811
Relation
serif::geometry::surface::detail::PressureSurfaceDescription<

EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 795
relation_catalog_contains_relation_v
serif::eos::relations, 112
relation_contains_input_v
serif::eos::relations, 112
relation_count
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 480
serif::discretization::domain::schema::validation::SchemaValidationResult, 909
relation_input
serif::geometry::surface::detail, 146
relationIndex
serif::discretization::domain::schema::validation::SchemaRelationValidationResult, 907
relationName
serif::discretization::domain::schema::validation::SchemaRelationValidationResult, 907
relationResults
serif::discretization::domain::schema::validation::SchemaValidationResult, 910
Relations
serif::eos::models::Polytrope, 767
relations
serif::eos::runtime::EOSView, 508
relations_type
serif::discretization::domain::schema::DomainSchema< DomainIDs, BoundaryIDs, Relations >, 478
RelationType
serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >, 905
RelationValidationFailure
serif::discretization::domain::relation::validation, 87
serif::discretization::domain::schema::validation, 90
Report
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 367
report
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 370
report_contribution_preparation
serif::operators::backend::mfem, 186
report_unused_overrides
serif::discretization::quadrature::QuadratureValidationOptions, 811
RequestedResults
serif::utils::dependencies, 198
require_explicit_base_order
serif::discretization::quadrature::QuadratureValidationOptions, 811
require_explicit_mfem_rule
serif::discretization::quadrature::QuadratureValidationOptions, 811
required
serif::utils::dependencies::detail::Graph< Computations >, 606
RESET
serif::utils::misc::terminal, 217
reset_from
serif::discretization::backend::mfem::BasisEvaluation, 297
serif::operators::backend::mfem::BasisEvaluation, 304
residual
serif::equations::EquationValidationResult, 762
serif::operators::backend::mfem::NodalData< EquationT, SchemaT >, 714
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >, 1011
residual_binding
serif::operators::backend::mfem::BasisBindings< ColumnCount >, 291
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >, 716
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >, 1013
residual_offset
serif::equations::AngularMomentum, 267
serif::equations::MassNormalization, 695
serif::operators::backend::mfem::ContributionData< OperandCount >, 391
ResidualBinding
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >, 715
ResidualQuantity
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 795
resolution
serif::discretization::quadrature::backend::mfem::MFEMRule, 708
resolve
serif::discretization::quadrature::backend::mfem::MFEMResolver, 707
serif::discretization::quadrature::Policy, 766
resolve_ansi_color
serif::utils::misc::terminal, 217
resolve_operand_order
serif::discretization::quadrature::backend::mfem, 106
ResolvedBindings
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true >, 873

ResolvedTargets
serif::utils::dependencies, 198
ResolverFor
serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > >, 325
serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... > >, 465
resolves_carrier
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 796
Resource
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >, 877
resource
serif::utils::dependencies::DependencyContext< ResourceTs >, 414
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >, 417
ResourceBindings
serif::utils::dependencies::DependencyContext< ResourceTs >, 413
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >, 416
Resources
serif::utils::dependencies::DependencyContext< ResourceTs >, 414
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >, 416
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >, 375
Result
serif::equations::GravityForce, 613
serif::equations::RotationalForce, 890
serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >, 719
result
serif::discretization::domain::schema::validation::SchemaRelationValidationResult, 907
ResultMask
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 367
Results
serif::utils::dependencies::ComputationGraph< Computations >, 377
results
serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > >, 373
reuse_unchanged
serif::utils::dependencies, 198
reused
serif::utils::dependencies, 199
reused_count
serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >, 782
reused_elements
serif::operators::backend::mfem::NumericalWork, 724
RigidRotation
serif::equations::RigidRotation, 883
rollection<
serif::discretization::quadrature::backend::mfem::MFEMQuadratu 706
serif::discretization::quadrature::Query, 821
role_control
serif::discretization::quadrature::RuleSet, 893
roles
serif::discretization::quadrature::QuadratureOptions, 808
RotationalForce
serif::equations::RotationalForce, 890
row_offsets
serif::geometry::deformation::RadialInterpolation, 828
RowBinding
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >, 1013
Rows
serif::equations::EquationRegistry< EquationTs >, 517
serif::operators::EquationRegistry< EquationTs >, 518
rule
serif::operators::backend::mfem::ElementBasis< OperandCount >, 485
runtime_failure
serif::eos::runtime::EOSView, 508
RuntimeCatalogQuantityTuple
serif::eos::runtime, 114
RuntimeEOSValidationResult, serif::eos::runtime, 114
RuntimeEvaluateFunction
serif::eos::runtime::EOSView, 507
runtimeEvaluationFailure
serif::eos::runtime, 114
RuntimePartialDerivativeFunction
serif::eos::runtime::EOSView, 507
runtimeQuantityIdentifierIsUnambiguous
serif::eos::runtime, 115
runtimeQuantityIdentifiersAreCompatible
serif::eos::runtime, 115
runtimeQuantityIdentifiersAreUnambiguous
serif::eos::runtime, 115
runtimeRelationDescriptors

serif::eos::runtime, 115
runtimeRelationMatches
serif::eos::runtime, 115

s_noElements
serif::discretization::domain::mesh::MeshTopology, 701
SameType
serif::utils::misc::std_helper, 215
sample
serif::discretization::backend::mfem::BasisEvaluation, 297
serif::geometry::surface::SurfaceFailure, 946
serif::operators::backend::mfem::BasisEvaluation, 304
sample_basis
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 731
sample_contribution_basis
serif::operators::backend::mfem, 186
sample_contribution_operand
serif::operators::backend::mfem, 187
sample_element_basis
serif::operators::backend::mfem, 187
sample_harmonic_basis
serif::geometry::deformation::detail, 136
sample_reference_positions
serif::geometry::deformation::backend::mfem, 131
sample_reference_surface
serif::geometry::deformation::backend::mfem, 131
sampled_operands
serif::operators::backend::mfem::NumericalWork, 724
sampled_point_count
serif::geometry::deformation::GeometryStepLimit, 593
SamplesFor
serif::operators::backend::mfem, 171
scalar_dof_count
serif::geometry::deformation::RadialInterpolation, 828
scalar_dofs
serif::geometry::deformation::ReferenceSurface, 839
scalar_space
serif::geometry::deformation::backend::mfem::ReferenceDiscretization, 831
scale
serif::mapping::compactification::KelvinCompactification::RadialFactors, 826
scale_derivative
serif::mapping::compactification::KelvinCompactification::Radial 826
ScaledMechanical
serif::operators, 158
ScaledPoisson
serif::operators, 158
scatter
serif::discretization::backend::mfem::DofMap, 442
Schema
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >, 966
serif::operators, 159
serif::operators::backend::mfem, 171
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >, 1011
second
serif::discretization::backend::mfem::TensorWorkspace, 960
secondDomainID
serif::discretization::domain::relation::validation::RelationValidati 455
serif::discretization::domain::schema::validation::RelationValidati 456
secondElementID
serif::discretization::domain::mesh::FaceElements, 529
serif::discretization::domain::relation::validation::RelationValidati 455
serif::discretization::domain::schema::validation::RelationValidati 456
select
serif::utils::dependencies::DependencyContext< ResourceTs >, 414, 415
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >, 332
select_boundary_dofs
serif::discretization::backend::mfem, 70
select_face_element_integration_point
serif::mapping::DomainMapper, 473
select_face_element_transformation
serif::mapping::DomainMapper, 473
select_nodal_values
serif::operators::backend::mfem, 187
select_point_dofs
serif::discretization::backend::mfem, 71
select_surface_scalar_dofs
serif::geometry::deformation::backend::mfem, 131
selected_rows
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >, 716
serif::operators::backend::mfem::SelectedNodalValues, 911

sequence
serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot, 836
serif::geometry::deformation::backend::mfem::ReferenceSpatialSnapshot, 838
serif::mapping::DMGeometryCache, 432
SERiF, 1
serif, 57
serif::dimensions, 57
AccelerationValue, 62
AngularMomentumValue, 62
AngularVelocityValue, 62
AreaValue, 62
ChemicalPotentialValue, 62
DensityValue, 62
DimensionlessValue, 62
DynamicViscosityValue, 62
EnergyDensityValue, 63
EnergyValue, 63
EntropyValue, 63
ForceValue, 63
FrequencyValue, 63
GravitationalPotentialValue, 63
InternalEnergyValue, 63
KinematicViscosityValue, 64
LengthValue, 64
LuminosityValue, 64
MagneticFluxDensityValue, 64
MassFlowRateValue, 64
MassValue, 64
MomentOfInertiaValue, 64
MomentumValue, 65
NeitherConstNorVolatile, 65
NeitherConstNorVolatileReference, 65
NumberDensityValue, 65
OpacityValue, 65
PowerValue, 65
PressureValue, 65
QuantityOfT, 66
SpecificEnergyValue, 66
SpecificEnthalpyValue, 66
SpecificEntropyValue, 66
SpecificInternalEnergyValue, 66
TemperatureValue, 66
TimeValue, 66
TorqueValue, 67
VelocityValue, 67
VolumeValue, 67
serif::dimensions::Acceleration, 259
identifier, 260
serif::dimensions::AngularMomentum, 263
identifier, 264
serif::dimensions::AngularVelocity, 269
identifier, 270
serif::dimensions::Area, 273
MeshSnapshot, 271
serif::dimensions::ChemicalPotential, 343
SpatialSnapshot, 341
serif::dimensions::Dimensionless, 418
identifier, 419
serif::dimensions::DynamicViscosity, 482
identifier, 483
serif::dimensions::Energy, 493
identifier, 494
serif::dimensions::EnergyDensity, 494
identifier, 495
serif::dimensions::Entropy, 498
identifier, 499
serif::dimensions::Force, 538
identifier, 539
serif::dimensions::Frequency, 582
identifier, 584
serif::dimensions::GravitationalPotential, 607
identifier, 609
serif::dimensions::InternalEnergy, 639
identifier, 640
serif::dimensions::IsQuantityValue< Candidate >, 650
serif::dimensions::IsQuantityValue< QuantityValue< Quantity > >, 651
serif::dimensions::KinematicViscosity, 668
identifier, 670
serif::dimensions::Length, 670
identifier, 671
serif::dimensions::Luminosity, 673
identifier, 674
serif::dimensions::MagneticFluxDensity, 675
identifier, 676
serif::dimensions::Mass, 679
identifier, 680
serif::dimensions::MassDensity, 690
identifier, 691
serif::dimensions::MassFlowRate, 691
identifier, 692
serif::dimensions::MomentOfInertia, 709
identifier, 710
serif::dimensions::Momentum, 710
identifier, 711
serif::dimensions::NumberDensity, 722
identifier, 723
serif::dimensions::Opacity, 725
identifier, 726
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >, 749
m_value, 755
operator<=>, 754, 755
operator+, 752
operator-, 753

operator/, 753
operator∗, 752
PartialDerivative, 751
value, 751
serif::dimensions::PhysicalQuantity, 758
serif::dimensions::PhysicalQuantityType, 225
serif::dimensions::Power, 776
identifier, 777
serif::dimensions::Pressure, 784
identifier, 785
serif::dimensions::QuantityAt< Index, Quantities >, 812
serif::dimensions::QuantityAt< Index, QuantityList< Quantities... > >, 812
Type, 812
serif::dimensions::QuantityList< Quantities >, 812
serif::dimensions::QuantityOf< Candidate >, 812
serif::dimensions::QuantityOf< QuantityValue< Quantity > >, 813
Type, 813
serif::dimensions::QuantityValue< Quantity >, 813
m_value, 820
operator<=>, 818, 819
operator+, 816
operator-, 816
operator/, 816
operator==, 819
operator∗, 815
QuantityValue, 814
value, 815
serif::dimensions::QuantityValueType, 226
serif::dimensions::runtime, 67
thermodynamicQuantityID, 68
serif::dimensions::runtime::RuntimeIdentifiedThermodynamicQuantity, 226
serif::dimensions::runtime::RuntimeQuantityValue, 899
id, 900
value, 900
serif::dimensions::runtime::RuntimeRelationDescriptor, 900
hasPartialDerivative, 900
inputQuantities, 901
outputQuantity, 901
partialDerivativeMask, 901
serif::dimensions::runtime::ThermodynamicQuantityID, 973
m_name, 975
name, 975
operator==, 975
ThermodynamicQuantityID, 974
serif::dimensions::SpecificEnergy, 926
identifier, 927
serif::dimensions::SpecificEnthalpy, 928
identifier, 929
serif::dimensions::SpecificEntropy, 929
identifier, 930
serif::dimensions::SpecificInternalEnergy, 931
identifier, 932
serif::dimensions::Temperature, 948
identifier, 949
serif::dimensions::ThermodynamicQuantity, 971
serif::dimensions::ThermodynamicQuantityType, 226
serif::dimensions::ThermodynamicQuantityValueType, 227
serif::dimensions::Time, 975
identifier, 976
serif::dimensions::Torque, 977
identifier, 978
serif::dimensions::Velocity, 1004
identifier, 1005
serif::dimensions::Volume, 1005
identifier, 1006
serif::dimensions::WithRespectTo< Quantity >, 1024
serif::discretization, 68
serif::discretization::backend, 68
serif::discretization::backend::mfem, 69
automatic, 70
ContractionMode, 70
dense, 70
divergence, 70
gradient, 70
ReferenceOperation, 70
select_boundary_dofs, 70
select_point_dofs, 71
value, 70
value_and_gradient, 70
serif::discretization::backend::mfem::BasisEvaluation, 292
evaluate, 296
height, 296
integrate, 296
m_components, 298
m_dense, 298
m_mapping, 299
m_points, 299
m_tensor, 299
m_width, 299
map_point, 296
reset_from, 297
sample, 297
set_constant, 298
uses_tensor, 298
width, 298
serif::discretization::backend::mfem::BasisWorkspace, 307
reference_values, 307
tensor, 307

serif::discretization::backend::mfem::CoefficientTracking, 347
capture, 348, 349
identity, 349
matches, 349
serif::discretization::backend::mfem::DofMap, 437
~DofMap, 440
add_element, 440
DofMap, 440
element_values, 441
full_size, 441
gather, 441
m_attributes, 444
m_element_support, 444
m_full_to_reduced, 444
m_mesh, 444
m_mesh_sequence, 444
m_reduced_to_full, 444
m_space, 444
m_space_sequence, 445
mesh, 441
operator=, 442
reduced_dof, 442
scatter, 442
size, 443
space, 443
supports, 443
validate, 443
validate_values, 443
serif::discretization::backend::mfem::DofSelection, 445
binding, 447
boundary_attributes, 447
DofSelection, 446
m_binding, 448
m_boundary_attributes, 448
m_dofs, 448
m_nodes_sequence, 449
reduced_dofs, 447
size, 447
tracks_coordinates, 448
validate, 448
serif::discretization::backend::mfem::TensorBasis, 949
agrees_with, 951
apply, 952
components, 952
contract_axis, 953
dofs, 953
m_components, 954
m_dofs, 954
m_points, 955
m_products, 955
make, 953
points, 954
serif::discretization::backend::mfem::TensorBasis::Product, 799
dofs, 799
factors, 799
output, 800
serif::discretization::backend::mfem::TensorBasisCache, 955
clear, 956
get, 956
m_entries, 957
size, 957
serif::discretization::backend::mfem::TensorBasisCache::Entry, 499
basis, 500
copies, 500
element, 500
operation, 500
points, 500
serif::discretization::backend::mfem::TensorWorkspace, 959
coefficients, 960
first, 960
second, 960
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >, 964
checked_mesh, 967
element_support, 967
Schema, 966
Term, 966
TermBinding, 966, 967
serif::discretization::blocks, 72
DYNAMIC_BLOCK_SIZE, 76
field_descriptor, 76
FieldOf, 75
FunctionalSpaceOf, 75
QuantityOf, 75
SupportOf, 76
term_descriptor, 77
TermsOf, 76
serif::discretization::blocks::barotropic_constant, 280
mass_normalization_term, 282
name, 282
Terms, 281
serif::discretization::blocks::barotropic_constant::mass_normalization 688
name, 689
Quantity, 689
Support, 689
symbol, 689
serif::discretization::blocks::block, 310
serif::discretization::blocks::central_density, 337
border_term, 339
name, 339
Terms, 338

serif::discretization::blocks::central_density::border, 312
name, 313
Quantity, 313
Support, 313
symbol, 313
serif::discretization::blocks::density, 409
mass_density_term, 410
name, 410
Terms, 410
serif::discretization::blocks::density::mass, 681
FunctionalSpace, 683
name, 683
Quantity, 683
Support, 683
symbol, 683
serif::discretization::blocks::density::mass::residual, 854
serif::discretization::blocks::density::mass::value, 983
serif::discretization::blocks::displacement, 424
name, 425
Terms, 425
vector_term, 425
serif::discretization::blocks::displacement::vector, 1000
FunctionalSpace, 1002
name, 1003
Quantity, 1002
Support, 1002
symbol, 1003
serif::discretization::blocks::displacement::vector::residual, 856
serif::discretization::blocks::displacement::vector::value, 984
serif::discretization::blocks::enthalpy, 496
name, 497
specific_term, 497
Terms, 497
serif::discretization::blocks::enthalpy::specific, 923
FunctionalSpace, 925
name, 925
Quantity, 925
Support, 925
symbol, 925
serif::discretization::blocks::enthalpy::specific::residual, 857
serif::discretization::blocks::enthalpy::specific::value, 986
serif::discretization::blocks::Field, 227
serif::discretization::blocks::field, 533
serif::discretization::blocks::FieldDescriptor, 534
name, 535
operator==, 535
serif::discretization::blocks::generated_residual_block< GeneratedResidual >, 587
GeneratedType, 589
static_block_size, 589
serif::discretization::blocks::generated_value_block< GeneratedValue >, 589
GeneratedType, 591
static_block_size, 591
serif::discretization::blocks::global_scalar, 595
serif::discretization::blocks::global_scalar::residual, 859
static_block_size, 860
serif::discretization::blocks::global_scalar::value, 987
static_block_size, 989
serif::discretization::blocks::GlobalScalarTerm, 228
serif::discretization::blocks::gravity, 609
gradient_term, 611
name, 611
potential_term, 611
Terms, 610
serif::discretization::blocks::gravity::gradient, 596
FunctionalSpace, 598
name, 598
Quantity, 598
Support, 598
symbol, 598
serif::discretization::blocks::gravity::gradient::residual, 861
serif::discretization::blocks::gravity::gradient::value, 989
serif::discretization::blocks::gravity::potential, 770
FunctionalSpace, 772
name, 773
Quantity, 772
Support, 773
symbol, 773
serif::discretization::blocks::gravity::potential::residual, 862
serif::discretization::blocks::gravity::potential::value, 991
serif::discretization::blocks::HasFunctionalSpace, 228
serif::discretization::blocks::HasQuantity, 228
serif::discretization::blocks::HasResidualBlock, 229
serif::discretization::blocks::HasSupport, 229
serif::discretization::blocks::HasValueBlock, 230
serif::discretization::blocks::IsGlobalScalarTerm< T >, 644
serif::discretization::blocks::IsPhysicalSpatialTerm< T >, 648
serif::discretization::blocks::IsSpatialTerm< T >, 652
serif::discretization::blocks::PhysicalSpatialTerm, 230
serif::discretization::blocks::residual_block< index_value >, 865
index, 867

operator int, 867
serif::discretization::blocks::residual_block_base, 868
static_block_size, 869
serif::discretization::blocks::rotation, 886
angular_velocity_term, 887
name, 887
Terms, 887
serif::discretization::blocks::rotation::angular_velocity, 261
name, 262
Quantity, 262
Support, 262
symbol, 262
serif::discretization::blocks::SpatialTerm, 231
serif::discretization::blocks::SpatialTermTraits< TermT >, 921
family, 922
FunctionalSpace, 922
order, 922
rank, 922
Support, 922
serif::discretization::blocks::surface_deformation, 936
coefficients_term, 938
name, 938
shape_equilibrium_term, 938
Terms, 937
serif::discretization::blocks::surface_deformation::coefficients, 345
name, 347
serif::discretization::blocks::surface_deformation::coefficients::value, 992
serif::discretization::blocks::surface_deformation::shape_equilibrium, 912
name, 914
serif::discretization::blocks::surface_deformation::shape_equilibrium::value, 864
serif::discretization::blocks::Term, 231
serif::discretization::blocks::term< FieldT >, 961
field_type, 962
serif::discretization::blocks::term_base, 962
serif::discretization::blocks::TermDescriptor, 968
field, 969
name, 969
operator==, 969
serif::discretization::blocks::value_block< index_value >, 994
index, 996
operator int, 996
serif::discretization::blocks::value_block_base, 996
static_block_size, 998
serif::discretization::domain, 77
AIsBaseClassOfB, 79
AllDomains, 78
CountVariadicArguments, 78
is_domain_set_v, 79
is_domain_set_v< DomainSet< DomainTs... > >, 79
StellarDomains, 78
serif::discretization::domain::Boundary, 315
serif::discretization::domain::BoundaryIDResolver< BoundarySet< BoundaryTs... >, ids::lists::BoundaryIDList< BoundaryIDTs... > >, 325
contains_id, 325
registered, 325
ResolverFor, 325
serif::discretization::domain::BoundaryIDResolver< BoundaryT, BoundaryIDListT >, 324
serif::discretization::domain::BoundaryIDResolver< BoundaryT, ids::lists::BoundaryIDList< BoundaryIDTs... > >, 326
contains_id, 326
id, 326
registered, 326
serif::discretization::domain::BoundaryIsIdentifiedBy, 231
serif::discretization::domain::BoundarySet< BoundaryTs >, 327
count, 327
serif::discretization::domain::CoreDomain, 397
name, 398
serif::discretization::domain::Domain, 449
serif::discretization::domain::DomainIDResolver< DomainSet< DomainTs... >, ids::lists::DomainIDList< DomainIDTs... > >, 465
contains_id, 465
registered, 466
ResolverFor, 465
serif::discretization::domain::DomainIDResolver< DomainT, DomainIDListT >, 465
serif::discretization::domain::DomainIDResolver< DomainT, ids::lists::DomainIDList< DomainIDTs... > >, 466
contains_id, 466
id, 466
registered, 467
serif::discretization::domain::DomainIsIdentifiedBy, 232
serif::discretization::domain::DomainOrSet, 475
DomainOrSet, 475
domains, 476
m_domains, 476
serif::discretization::domain::DomainOrSetList< DomainTs >, 476
count, 476
serif::discretization::domain::DomainSet< DomainTs >, 481
count, 481

names, 481
serif::discretization::domain::EnvelopeDomain, 503
name, 504
serif::discretization::domain::ids, 79
is_boundary_id_v, 80
is_boundary_id_v< BoundaryID< BoundaryT, IdentifierValue > >, 80
is_domain_id_v, 81
is_domain_id_v< DomainID< DomainT, IdentifierValue > >, 81
serif::discretization::domain::ids::BoundaryDescriptor, 318
ID, 319
name, 319
serif::discretization::domain::ids::BoundaryID< BoundaryT, IdentifierValue >, 319
boundary_type, 320
ID, 320
serif::discretization::domain::ids::DomainDescriptor, 458
ID, 459
name, 459
serif::discretization::domain::ids::DomainID< DomainT, IdentifierValue >, 460
domain_type, 461
ID, 461
serif::discretization::domain::ids::IsBoundaryID, 232
serif::discretization::domain::ids::IsDomainID, 232
serif::discretization::domain::ids::IsDomainOrBoundaryID, 233
serif::discretization::domain::ids::lists, 81
CountVariadicArguments, 82
is_boundary_id_list_v, 82
is_boundary_id_list_v< BoundaryIDList< BoundaryIDTs... > >, 82
is_domain_id_list_v, 83
is_domain_id_list_v< DomainIDList< DomainIDTs... > >, 83
serif::discretization::domain::ids::lists::BoundaryIDList< BoundaryIDTs >, 322
count, 323
descriptors, 323
serif::discretization::domain::ids::lists::DomainIDList< DomainIDTs >, 462
count, 463
descriptors, 463
serif::discretization::domain::ids::lists::IsBoundaryIDList, 233
serif::discretization::domain::ids::lists::IsDomainIDList, 234
serif::discretization::domain::ids::validation, 83
all_unique_values, 84
serif::discretization::domain::ids::validation::AreDistinctBoundaryIDs, 234
serif::discretization::domain::ids::validation::AreDistinctDomainIDs, 235
serif::discretization::domain::ids::validation::HaveUniqueBoundaries, 235
serif::discretization::domain::ids::validation::HaveUniqueBoundaryIDs, 236
serif::discretization::domain::ids::validation::HaveUniqueDomainIDs, 236
serif::discretization::domain::ids::validation::HaveUniqueDomains, 237
serif::discretization::domain::InfinitySurfaceBoundary, 631
name, 632
serif::discretization::domain::IsBoundary, 237
serif::discretization::domain::IsDomain, 237
serif::discretization::domain::IsDomainOrSet, 238
serif::discretization::domain::IsDomainSet, 238
serif::discretization::domain::mesh, 84
serif::discretization::domain::mesh::FaceElements, 529
firstElementID, 529
secondElementID, 529
serif::discretization::domain::mesh::MeshTopology, 698
boundary_element_boundary_id, 699
boundary_element_count, 699
boundary_element_face, 699
boundary_elements_on_face, 700
element_count, 700
element_domain_id, 700
element_neighbors, 700
face_count, 700
face_elements, 700
m_boundaryElementBoundaryIDs, 700
m_boundaryElementFaceIDs, 700
m_boundaryElementsByFace, 701
m_elementDomainIDs, 701
m_elementNeighbors, 701
m_faceElements, 701
MeshTopology, 699
s_noElements, 701
serif::discretization::domain::relation, 84
AIsBaseClassOfB, 85
CountVariadicArguments, 85
serif::discretization::domain::relation::DomainBoundary< BoundaryT, DomainTs >, 450
boundary_type, 451
domain_count, 451
domain_types, 451
name, 451
serif::discretization::domain::relation::DomainRelation, 477
serif::discretization::domain::relation::FullyConnected< A >, 584

domain_type, 585
name, 585
serif::discretization::domain::relation::Inscribed< A, B >, 633
inner_type, 634
name, 635
outer_type, 634
serif::discretization::domain::relation::IsRelation, 238
serif::discretization::domain::relation::lists, 85
CountVariadicArguments, 86
is_relation_list_v, 86
is_relation_list_v< RelationList< RelationTs... > >, 86
serif::discretization::domain::relation::lists::IsRelationList, 238
serif::discretization::domain::relation::lists::RelationList< RelationTs >, 844
count, 844
serif::discretization::domain::relation::validation, 86
DomainAbsent, 87
DomainBoundaryAbsent, 87
DomainBoundaryExpectedFaceHasWrongID, 87
DomainBoundaryExpectedFaceIsUntagged, 87
DomainBoundaryTaggedFaceHasWrongTopology, 87
DomainBoundaryTaggedFaceTouchesUnexpectedDomain, 87
DomainDisconnected, 87
InnerDomainAbsent, 87
InnerDomainHasNoBoundary, 87
InnerDomainTouchesMeshBoundary, 87
InnerDomainTouchesUnexpectedDomain, 87
None, 87
OuterDomainAbsent, 87
RelationValidationFailure, 87
serif::discretization::domain::relation::validation::HaveValidRelationBoundaries, 239
serif::discretization::domain::relation::validation::RelationIsRegistered< DomainIDsT, BoundaryIDsT, lists::RelationList< RelationTs... > >, 846
serif::discretization::domain::relation::validation::RelationsUseRegisteredDomains< DomainIDsT, BoundaryIDsT, RelationListT >, 846
serif::discretization::domain::relation::validation::RelationUsesRegisteredDomains< DomainBoundary< BoundaryT, DomainTs... >, DomainIDsT, BoundaryIDsT >, 847
serif::discretization::domain::relation::validation::RelationUsesRegisteredDomains< FullyConnected< DomainT >, DomainIDsT, BoundaryIDsT >, 848
serif::discretization::domain::relation::validation::RelationUsesRegisteredDomains< Inscribed< InnerDomainT, OuterDomainT >, DomainIDsT, BoundaryIDsT >, 849
serif::discretization::domain::relation::validation::RelationUsesRegisteredDomains< RelationT, DomainIDsT, BoundaryIDsT >, 847
serif::discretization::domain::relation::validation::RelationValidationR 849
connectedDiagnostics, 850
domainBoundaryDiagnostics, 850
failure, 850
inscribedDiagnostics, 851
operator bool, 850
valid, 850
serif::discretization::domain::relation::validation::RelationValidationR 386
domainElementCount, 386
elementID, 386
visitedElementCount, 386
serif::discretization::domain::relation::validation::RelationValidationR 454
actualBoundaryID, 454
boundaryElementID, 454
expectedBoundaryID, 454
faceID, 454
firstDomainID, 454
firstElementID, 454
secondDomainID, 455
secondElementID, 455
serif::discretization::domain::relation::validation::RelationValidationR 637
adjacentDomainID, 637
adjacentElementID, 637
faceID, 637
innerElementID, 637
serif::discretization::domain::schema, 87
CoreEnvelopeVacuumDomainSchema, 89
is_schema_v, 89
is_schema_v< DomainSchema< DomainIDs, BoundaryIDs, Relations > >, 89
make_attribute_marker, 89
serif::discretization::domain::schema::BoundaryID< BoundaryT, IdentifierValue >, 321
boundary_type, 321
serif::discretization::domain::schema::BoundaryIDList< BoundaryIDTs >, 323
serif::discretization::domain::schema::DomainID< DomainT, IdentifierValue >, 461
domain_type, 462
ID, 462
serif::discretization::domain::schema::DomainIDList< DomainIDTs >, 463
count, 464
descriptors, 464
serif::discretization::domain::schema::DomainSchema<

DomainIDs, BoundaryIDs, Relations >, 477
boundary_descriptors, 479
boundary_id, 479
boundary_id_count, 480
boundary_id_matches, 479
boundary_ids_type, 478
contains_boundary, 479
contains_domain, 479
domain_descriptors, 479
domain_id, 479
domain_id_belongs_to, 480
domain_id_count, 480
domain_ids_type, 478
relation_count, 480
relations_type, 478
serif::discretization::domain::schema::IsSchema, 239
serif::discretization::domain::schema::RelationList< RelationTs >, 844
count, 844
serif::discretization::domain::schema::validation, 90
RelationValidationFailure, 90
validate_schema, 91
serif::discretization::domain::schema::validation::DomainBoundaryDescriptor< BoundaryT, DomainTs >, 452
boundary_type, 453
domain_count, 453
domain_types, 453
name, 453
serif::discretization::domain::schema::validation::FullyConnected< A >, 586
domain_type, 587
name, 587
serif::discretization::domain::schema::validation::Inscribed< A, B >, 635
inner_type, 636
name, 637
outer_type, 636
serif::discretization::domain::schema::validation::RelationValidationResult, 851
connectedDiagnostics, 852
domainBoundaryDiagnostics, 852
failure, 852
inscribedDiagnostics, 852
operator bool, 851
valid, 851
serif::discretization::domain::schema::validation::RelationValidationResult::ConnectedDiagnostics, 386
domainElementCount, 387
elementID, 387
visitedElementCount, 387
serif::discretization::domain::schema::validation::RelationValidationResult::DomainBoundaryDiagnostics, 455
actualBoundaryID, 455
boundaryElementID, 455
expectedBoundaryID, 456
faceID, 456
firstDomainID, 456
firstElementID, 456
secondDomainID, 456
secondElementID, 456
serif::discretization::domain::schema::validation::RelationValidationResult::InscribedDiagnostics, 638
adjacentDomainID, 638
adjacentElementID, 638
faceID, 638
innerElementID, 638
serif::discretization::domain::schema::validation::RelationValidator< DomainBoundary< BoundaryT, DomainTs... > >, 852
validate, 853
serif::discretization::domain::schema::validation::RelationValidator< FullyConnected< DomainT > >, 853
validate, 853
serif::discretization::domain::schema::validation::RelationValidator< Inscribed< InnerT, OuterT > >, 854
validate, 854
serif::discretization::domain::schema::validation::RelationValidator< RelationT >, 852
serif::discretization::domain::schema::validation::SchemaRelationValidationResult, 906
operator bool, 907
relationIndex, 907
relationName, 907
result, 907
valid, 907
serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, relation::lists::RelationList< RelationTs... > >, 908
validate, 908
serif::discretization::domain::schema::validation::SchemaRelationValidator< SchemaT, RelationListT >, 908
serif::discretization::domain::schema::validation::SchemaValidationResult, 908
failed_relation_count, 909
first_failed_relation_index, 909
operator bool, 909
passed_relation_count, 909
relation_count, 909
relationResults, 910
valid, 909
serif::discretization::domain::StellarSurfaceBoundary, 935
name, 936
serif::discretization::domain::VacuumDomain, 979
name, 980
serif::discretization::forms, 91
component, 95
component_count, 96

derivative_component, 95
form_descriptor, 96
MatrixValue, 94
OperandValue, 94
PartialDerivative, 94
VectorValue, 94
serif::discretization::forms::boundary_flux< TrialT, TestT >, 316
name, 318
operand_types, 317
test_type, 317
trial_type, 317
serif::discretization::forms::divergence< TrialT, TestT >, 428
name, 430
operand_types, 430
test_type, 430
trial_type, 430
serif::discretization::forms::force< TrialT, TestT >, 539
name, 541
operand_types, 541
test_type, 541
trial_type, 541
serif::discretization::forms::Form, 239
serif::discretization::forms::form, 542
serif::discretization::forms::FormDescriptor, 571
name, 572
operator==, 572
test, 572
trial, 573
serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > >, 575
test_order_reduction, 575
trial_order_reduction, 575
serif::discretization::forms::FormTraits< divergence< TrialT, TestT > >, 576
test_order_reduction, 576
trial_order_reduction, 577
serif::discretization::forms::FormTraits< force< TrialT, TestT > >, 577
test_order_reduction, 578
trial_order_reduction, 578
serif::discretization::forms::FormTraits< FormT >, 574
test_order_reduction, 574
trial_order_reduction, 574
serif::discretization::forms::FormTraits< mass< TrialT, TestT > >, 578
test_order_reduction, 579
trial_order_reduction, 579
serif::discretization::forms::FormTraits< projection< TrialT, TestT > >, 579
test_order_reduction, 581
trial_order_reduction, 581
serif::discretization::forms::FormTraits< source< TrialT, TestT > >, 581
test_order_reduction, 582
trial_order_reduction, 582
serif::discretization::forms::IsOperand, 240
serif::discretization::forms::IsOperation, 240
serif::discretization::forms::mass< TrialT, TestT >, 684
name, 687
operand_types, 687
test_type, 687
trial_type, 687
serif::discretization::forms::Operand< TermT, OperationT >, 727
operation_type, 727
term_type, 727
serif::discretization::forms::OperandResult< OperandT >, 733, 734
type, 735
serif::discretization::forms::Operation, 739
serif::discretization::forms::Operation::Curl, 401
serif::discretization::forms::Operation::Divergence, 428
serif::discretization::forms::Operation::Gradient, 596
serif::discretization::forms::Operation::NormalTrace, 721
serif::discretization::forms::Operation::Value, 982
serif::discretization::forms::OperationValue< 0, Operation::Gradient >, 740
type, 741
serif::discretization::forms::OperationValue< 0, Operation::Value >, 741
type, 741
serif::discretization::forms::OperationValue< 1, Operation::Curl >, 741
type, 742
serif::discretization::forms::OperationValue< 1, Operation::Divergence >, 742
type, 742
serif::discretization::forms::OperationValue< 1, Operation::NormalTrace >, 743
type, 743
serif::discretization::forms::OperationValue< 1, Operation::Value >, 743
type, 744
serif::discretization::forms::OperationValue< Rank, OperationT >, 740
serif::discretization::forms::projection< TrialT, TestT >, 801
name, 804
operand_types, 803
test_type, 803
trial_type, 803

serif::discretization::forms::source< TrialT, TestT >, 915
name, 917
operand_types, 916
test_type, 916
trial_type, 916
serif::discretization::functional_space, 96
Continuity, 98
Continuity2StringMap, 99
continuous, 98
contravariant_piola, 99
covariant_piola, 99
discontinuous, 98
Family, 98
Family2StringMap, 99
H1, 99
h1, 98
Hcurl, 100
hcurl, 98
Hdiv, 100
hdiv, 98
identity, 99
L2, 101
l2, 98
Mapping, 98
Mapping2StringMap, 101
none, 99
normal, 99
normal_continuous, 98
String2ContinuityMap, 101
String2FamilyMap, 102
String2MappingMap, 102
String2TraceMap, 102
tangential, 99
tangential_continuous, 98
Trace, 99
Trace2StringMap, 102
value, 99
serif::discretization::functional_space::DiscreteFunctionalSpace< Space, Order, R >, 421
functional_space, 422
order, 422
rank, 422
serif::discretization::functional_space::IsSpaceDescriptor, 241
serif::discretization::functional_space::Rank< R >, 829
value, 830
serif::discretization::functional_space::SpaceDescriptor, 918
continuity, 919
family, 919
mapping, 919
trace, 919
serif::discretization::quadrature, 102
affine, 103
convergence, 104
diagnostic, 104
discretization, 104
fast, 104
general, 103
kelvin, 103
make_query, 104
MappingKind, 103
Mode, 103
none, 103
preconditioner, 104
production, 104
projection, 104
QuadratureRole, 104
reference, 104
serif::discretization::quadrature::backend, 104
serif::discretization::quadrature::backend::mfem, 105
FunctionalSpaceTraits, 105
mfem_form_static_order, 106
resolve_operand_order, 106
set_integration_rule, 106
sum_dynamic_orders, 106
serif::discretization::quadrature::backend::mfem::ConfigurableIntegrat 241
serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSp functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > >, 423
order, 423
rank, 423
space, 423
serif::discretization::quadrature::backend::mfem::DiscreteFunctionalSp FunctionalSpaceT >, 422
serif::discretization::quadrature::backend::mfem::HasPointerSetIntegra 241
serif::discretization::quadrature::backend::mfem::HasReferenceSetInte 242
serif::discretization::quadrature::backend::mfem::HasSetIntRule, 242
serif::discretization::quadrature::backend::mfem::MFEMOperandListC OperandListT >, 703
serif::discretization::quadrature::backend::mfem::MFEMOperandListC utils::types::TypeList< OperandTs... > >, 703
value, 704
serif::discretization::quadrature::backend::mfem::MFEMQuadratureC 704
domain, 705
dynamic_orders, 705
geometry, 706
geometry_weight_order, 706
mapping, 706
role, 706

serif::discretization::quadrature::backend::mfem::MFEMResolver, 706
configure, 707
m_policy, 707
MFEMResolver, 707
resolve, 707
serif::discretization::quadrature::backend::mfem::MFEMRule, 708
integration_rule, 708
resolution, 708
serif::discretization::quadrature::FormRule, 573
control, 573
form, 573
serif::discretization::quadrature::Policy, 765
compute_base_order, 765
get_role_control, 765
m_ruleSet, 766
Policy, 765
resolve, 766
serif::discretization::quadrature::QuadratureManifestOptions, 804
enabled, 804
include_repeated_queries, 804
output_file, 805
serif::discretization::quadrature::QuadratureOptions, 805
center_of_mass, 806
centrifugal, 806
convergence_boosts, 806
density_projection, 806
eos_closure, 806
error_norm, 806
fallback_fixed_order, 807
global_boost, 807
gravitational_energy, 807
gravity_boundary, 807
gravity_divergence, 807
gravity_force, 807
gravity_hdiv_mass, 807
gravity_source, 807
hydrostatic_equilibrium, 807
isobaric_surface, 807
manifest, 808
mass_conservation, 808
mass_normalization, 808
mesh_extension, 808
mode, 808
pressure_force, 808
pressure_integral, 808
quadrupole, 808
roles, 808
validation, 808
virial, 809
serif::discretization::quadrature::QuadratureRoleOptions, 809
diagnostic, 810
discretization, 810
preconditioner, 810
projection, 810
serif::discretization::quadrature::QuadratureTermOptions, 810
additional_boost, 810
fixed_order, 810
serif::discretization::quadrature::QuadratureValidationOptions, 811
reject_negative_boosts, 811
report_unused_overrides, 811
require_explicit_base_order, 811
require_explicit_mfem_rule, 811
serif::discretization::quadrature::Query, 820
base_order, 820
coefficient_order, 820
domain, 821
form, 821
geometry_weight_order, 821
mapping, 821
role, 821
test_order, 821
test_order_reduction, 821
trial_order, 821
trial_order_reduction, 821
serif::discretization::quadrature::Resolution, 869
base_order, 870
boost, 870
order, 870
used_fixed_order, 870
serif::discretization::quadrature::RoleControls, 885
diagnostic, 885
discretization, 885
preconditioner, 886
projection, 886
serif::discretization::quadrature::RuleControl, 891
boost, 891
fixed_order, 891
serif::discretization::quadrature::RuleSet, 892
begin, 892
end, 892, 893
fallback_control, 893
find_form_control, 893
m_diagnosticControl, 894
m_discretizationControl, 894
m_fallbackControl, 894
m_formRules, 894
m_preconditionerControl, 894
m_projectionControl, 894
role_control, 893
set_fallback_control, 893

set_form_control, 893
set_role_control, 893
serif::eos, 106
evaluate, 107
partial_derivative, 107
serif::eos::EOSEvaluationError, 504
EOSEvaluationError, 505
serif::eos::models, 107
is_equation_of_state_model_v, 109
SERiFErrorCode, 108
supports_partial_derivative_v, 109
supports_relation_v, 109
validate_finite, 109
validate_greater_than, 109
validate_less_than, 109
validate_nonnegativity, 109
serif::eos::models::BarotropicClosureEOS, 242
serif::eos::models::EOSModel, 242
serif::eos::models::EOSSupportingPartialDerivative, 243
serif::eos::models::EOSSupportingRelation, 243
serif::eos::models::ImplementsPartialDerivative_t< EOS, RelationT, InputQuantity >, 625
serif::eos::models::ImplementsRelation_t< EOS, RelationT >, 626
serif::eos::models::ImplementsRelationCatalog_t< EOS, Output, Inputs >, 627
serif::eos::models::ImplementsRelationCatalog_t< EOS, relations::RelationCatalog< Relations... > >, 628
serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, std::void_t< typename std::remove_cvref_t< EOSCandidate >::Relations > >, 643
serif::eos::models::IsEquationOfStateModel_t< EOSCandidate, typename >, 642
serif::eos::models::Polytrope, 766
enthalpy_scale, 768
evaluate, 768
m_enthalpy_scale, 769
m_polytropic_constant, 769
m_polytropic_index, 769
partial_derivative, 768, 769
Polytrope, 767
polytropic_constant, 769
polytropic_index, 769
Relations, 767
serif::eos::models::PolytropeParameters, 770
K, 770
n, 770
serif::eos::models::PressureForceEOS, 243
serif::eos::models::StructureSeedEOS, 243
serif::eos::relations, 110
CountVariadicArguments, 112
DensityFromSpecificEnthalpy, 111
is_thermodynamic_relation_v, 112
is_valid_relation_catalog_v, 112
PressureFromDensity, 111
PressureFromSpecificEnthalpy, 111
relation_catalog_contains_relation_v, 112
relation_contains_input_v, 112
SpecificEnthalpyFromDensity, 111
SpecificEnthalpyFromPressure, 111
serif::eos::relations::CatalogContainingRelation, 243
serif::eos::relations::CatalogContainsRelation< Catalog, RelationT >, 333
serif::eos::relations::CatalogContainsRelation< RelationCatalog< Relations... >, RelationT >, 334
serif::eos::relations::IsThermodynamicRelation< Relation< Output, Inputs... > >, 654
serif::eos::relations::IsThermodynamicRelation< RelationCandidate >, 653
serif::eos::relations::IsValidRelationCatalog< CandidateCatalog >, 655
serif::eos::relations::IsValidRelationCatalog< RelationCatalog< Relations... > >, 656
serif::eos::relations::Relation< Output, Inputs >, 840
inputCount, 841
InputQuantities, 840
OutputQuantity, 840
serif::eos::relations::RelationCatalog< Relations >, 841
size, 841
serif::eos::relations::RelationContainingInput, 244
serif::eos::relations::RelationContainsInput< Relation< Output, Inputs... >, Quantity >, 843
serif::eos::relations::RelationContainsInput< RelationT, Quantity >, 842
serif::eos::relations::ValidRelationCatalog, 244
serif::eos::relations::ValidThermodynamicRelation, 244
serif::eos::runtime, 113
evaluateRuntimePartialDerivative, 114
evaluateRuntimeRelation, 114
RuntimeCatalogQuantityTuple, 114
RuntimeEOSAdapter, 114
runtimeEvaluationFailure, 114
runtimeQuantityIdentifierIsUnambiguous, 115
runtimeQuantityIdentifiersAreCompatible, 115
runtimeQuantityIdentifiersAreUnambiguous, 115
runtimeRelationDescriptors, 115
runtimeRelationMatches, 115
tryRuntimePartialDerivative, 115
serif::eos::runtime::EOSView, 506
EOSView, 507
find_relation, 508

m_evaluate, 509
m_object, 509
m_partialDerivative, 509
m_relations, 509
relations, 508
runtime_failure, 508
RuntimeEvaluateFunction, 507
RuntimePartialDerivativeFunction, 507
supports, 508
try_evaluate, 508
try_partial_derivative, 509
validate_relation_request, 509
serif::eos::runtime::HasRuntimeQuantityIdentifiers< relations::Relation< OutputT, InputTs... > >, 622
serif::eos::runtime::HasRuntimeQuantityIdentifiers< RelationT >, 621
serif::eos::runtime::RuntimeCatalogDispatch< EOS, CatalogT >, 894
serif::eos::runtime::RuntimeCatalogDispatch< EOS, relations::RelationCatalog< RelationTs... > >, 895
evaluate, 895
partial_derivative, 895
serif::eos::runtime::RuntimeCatalogIsSupported< CatalogT >, 896
serif::eos::runtime::RuntimeCatalogIsSupported< relations::RelationCatalog< RelationTs... > >, 897
serif::eos::runtime::RuntimeCatalogStorage< EOS, CatalogT >, 897
serif::eos::runtime::RuntimeCatalogStorage< EOS, relations::RelationCatalog< RelationTs... > >, 898
descriptors, 898
serif::eos::runtime::RuntimeEOSModel, 244
serif::eos::runtime::RuntimeRelationQuantities< relations::Relation< OutputT, InputTs... > >, 901
Type, 902
serif::eos::runtime::RuntimeRelationQuantities< RelationT >, 901
serif::eos::runtime::RuntimeRelationsAreSupported< QuantitiesAreIdentified, RelationTs >, 902
serif::eos::runtime::RuntimeRelationsAreSupported< true, RelationTs... >, 903
serif::eos::runtime::RuntimeRelationStorage< EOS, relations::Relation< OutputT, InputTs... > >, 904
descriptor, 905
inputQuantityIDs, 905
makePartialDerivativeMask, 905
partialDerivativeMask, 906
RelationType, 905
serif::eos::runtime::RuntimeRelationStorage< EOS, RelationT >, 904
serif::equations, 116
ContributionsOf, 117
SupportOf, 117
TestOperandOf, 117
serif::equations::AngularMomentum, 265
additional_order, 267
AngularMomentum, 267
AngularVelocity, 266
AngularVelocityValue, 266
Density, 266
DensityValue, 266
evaluate, 267
m_rotation, 268
m_target_momentum, 268
name, 268
Operands, 267
residual_offset, 267
validate_integral, 267
validation_integrand, 268
serif::equations::BarotropicClosure< EOST >, 282
additional_order, 284
BarotropicClosure, 284
Enthalpy, 283
evaluate, 284
m_enthalpy_power, 284
m_eos, 284
name, 284
Operands, 283
Test, 283
serif::equations::BarotropicHydrostaticEquilibrium, 285
additional_order, 287
AngularVelocity, 286
BarotropicHydrostaticEquilibrium, 286
Constant, 286
evaluate, 287
m_rotation, 287
name, 287
Operands, 286
Potential, 286
Test, 286
serif::equations::CenteringConstraint, 335
evaluate, 337
name, 337
Operands, 336
position_independent, 337
reference_position, 337
serif::equations::CentralDensityConstraint< EOST >, 339
affine, 342
Border, 341
CentralDensityConstraint, 341

density_from_enthalpy, 341
evaluate, 341
m_eos, 342
m_target_density, 342
m_target_enthalpy, 342
name, 342
Operands, 341
position_independent, 342
reference_position, 343
target_density, 341
target_enthalpy, 342
serif::equations::Contribution< TestOperandT, SupportT >, 387
Support, 388
Test, 388
TestOperand, 388
serif::equations::Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions())> >, 393
type, 394
serif::equations::Contributions< T, typename >, 393
type, 393
serif::equations::ContributionSupport< T, std::void_t< typename T::Support > >, 395
type, 395
serif::equations::ContributionSupport< T, typename >, 394
type, 395
serif::equations::ContributionTuple< std::tuple< Ts... >, std::void_t< decltype(std::tuple_cat(std::declval< typename Contributions< std::remove_cvref_t< Ts > >::type >()...))> >, 396
type, 396
serif::equations::ContributionTuple< TupleT, typename >, 395
type, 396
serif::equations::DisplacementForce< EOST >, 426
contributions, 427
DisplacementForce, 427
m_equation, 427
name, 427
Test, 427
serif::equations::Equation, 245
serif::equations::EquationDifference< LeftT, RightT >, 510
contributions, 511
EquationDifference, 511
m_equation, 511
name, 511
Test, 511
serif::equations::EquationPoint< OperandsT >, 515
fields, 516
position, 516
serif::equations::EquationRegistry< EquationTs >, 516
EquationRegistry, 517
Equations, 517
for_each, 517
get, 517
m_equations, 518
Rows, 517
serif::equations::EquationScale< EquationT >, 520
coefficient, 521
contributions, 521
EquationScale, 521
m_coefficient, 521
m_equation, 521
name, 522
Test, 521
serif::equations::EquationSum< FirstT, RestTs >, 522
contributions, 523
EquationSum, 523
m_equations, 524
name, 524
Test, 523
serif::equations::gravity_contributions, 117
Density, 118
Divergence, 118
Gradient, 118
Operand, 118
Potential, 118
Value, 118
serif::equations::gravity_contributions::DensitySource, 411
additional_order, 412
DensitySource, 412
evaluate, 412
four_pi_G, 412
Operands, 412
serif::equations::gravity_contributions::GradientDivergence, 599
additional_order, 600
evaluate, 600
Operands, 600
serif::equations::gravity_contributions::GradientValue, 603
additional_order, 604
evaluate, 604
Operands, 604
serif::equations::gravity_contributions::PotentialValue, 774
additional_order, 775
evaluate, 775
Operands, 775
serif::equations::GravityForce, 611
additional_order, 613
Density, 612

DensityValue, 612
evaluate, 613
Gradient, 612
GradientValue, 613
name, 613
Operands, 613
Result, 613
serif::equations::GravityGradientRelation, 614
contributions, 615
name, 615
Test, 614
serif::equations::MassNormalization, 693
additional_order, 694
Density, 694
DensityValue, 694
evaluate, 694
m_target_mass, 695
MassNormalization, 694
name, 695
Operands, 694
residual_offset, 695
serif::equations::OperandValues< OperandsT, OutputT >, 736
serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >, 737
count, 738
Entry, 737
get, 738
index, 738
Storage, 737
values, 738
serif::equations::PointEquation, 245
serif::equations::PointLinearization< OperandsT, ResultT >, 762
field_derivatives, 762
position_derivative, 762
residual, 762
serif::equations::PoissonBalance, 763
contributions, 764
m_four_pi_G, 764
name, 764
PoissonBalance, 764
Test, 764
serif::equations::PressureForce< EOST >, 787
additional_order, 789
Enthalpy, 788
EnthalpyValue, 788
evaluate, 789
m_enthalpy_power, 789
m_eos, 789
name, 790
Operands, 788
PressureForce, 789
serif::equations::PressureSurfaceConstraint< EOST >, 790
Enthalpy, 791
evaluate, 792
m_target_enthalpy, 792
m_target_pressure, 792
name, 792
Operands, 791
position_independent, 793
PressureSurfaceConstraint, 792
target_pressure, 792
serif::equations::RigidRotation, 882
angular_velocity, 883
center, 883
gradient, 884
hessian, 884
m_angular_velocity, 884
m_center, 884
m_hessian, 884
Matrix, 883
potential, 884
RigidRotation, 883
unit_rotation, 884
Vector, 883
serif::equations::RotationalForce, 888
additional_order, 890
AngularVelocity, 889
AngularVelocityValue, 889
Density, 889
DensityValue, 889
evaluate, 890
m_rotation, 891
name, 891
Operands, 890
Result, 890
RotationalForce, 890
serif::equations::TestOperandTraits< T, std::void_t< typename T::TestOperand > >, 970
type, 971
serif::equations::TestOperandTraits< T, typename >, 970
type, 970
serif::equations::ValidContributions< RowT, std::tuple< C... > >, 982
serif::equations::ValidContributions< RowT, TupleT >, 981
serif::geometry, 118
serif::geometry::deformation, 119
apply_jacobian, 122
apply_jacobian_transpose, 123
apply_pullback_derivative, 123
build_displacement, 124
build_surface_embedding, 124
compose_deformation, 124

DeformationFailureReason, 122
DeformationStatus, 122
DeformationWorkspace, 122
evaluate_displacement, 125
incompatible_discretization, 122
invalid_configuration, 122
invalid_geometry, 122
invalid_size, 122
is_linear_deformation, 126
make_deformation_computations, 125
make_deformation_graph, 125
make_deformation_workspace, 125
make_radial_extension, 125
non_finite, 122
unresolved_surface, 122
validate_deformation_values, 126
serif::geometry::deformation::backend, 126
serif::geometry::deformation::backend::mfem, 126
build_radial_interpolation, 128
deformation_discretization, 129
estimate_geometry_step_limit, 129
make_deformation_computations, 129
make_deformation_graph, 129
make_deformation_inputs, 130
make_surface_metric_computation, 130
make_volume_displacement_computations, 130, 131
sample_reference_positions, 131
sample_reference_surface, 131
select_surface_scalar_dofs, 131
surface_parameter_mass_diagonal, 131
validate_companion_numbering, 132
validate_scalar_reference_space, 132
serif::geometry::deformation::backend::mfem::DeformationValueTracking, 404
identity, 406
serif::geometry::deformation::backend::mfem::detail, 132
boundary_logical_radius, 133
compile_radial_reference, 133
locate_boundary_point, 133
logical_radius, 133
share_reference_result, 134
serif::geometry::deformation::backend::mfem::detail::LocatedBoundaryPoint, 672
boundary_element, 673
point, 673
serif::geometry::deformation::backend::mfem::GeometrySampleRule, 591
element, 592
integration_rule, 592
serif::geometry::deformation::backend::mfem::ReferenceDiscretization, 830
displacement_space, 831
infinity_attribute, 831
logical_mesh, 831
scalar_space, 831
stellar_attributes, 831
surface_attribute, 831
vacuum_attributes, 831
serif::geometry::deformation::backend::mfem::ReferenceDiscretizationS 832
infinity_attribute, 833
meshes, 833
operator==, 833
spaces, 833
stellar_attributes, 833
surface_attribute, 833
vacuum_attributes, 833
serif::geometry::deformation::backend::mfem::ReferenceDiscretizationT 834
capture, 834
identity, 834
matches, 834
serif::geometry::deformation::backend::mfem::ReferenceMeshSnapshot 835
boundary_attributes, 836
element_attributes, 836
identity, 836
nodes_sequence, 836
operator==, 836
sequence, 836
vertices, 836
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapsho 837
collection, 837
identity, 837
ordering, 837
sequence, 838
size, 838
vector_dimension, 838
serif::geometry::deformation::backend::mfem::volume_displacement_l 134
serif::geometry::deformation::backend::mfem::volume_displacement_l 343
serif::geometry::deformation::backend::mfem::volume_displacement_l 345
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 358
apply_jacobian, 360
apply_jacobian_transpose, 360
apply_pullback_derivative, 360
build_displacement, 361
compose_deformation, 362
ComposedDeformation, 360
extension_map, 361
input_size, 361

InputSpace, 360
is_linear, 362
m_extension, 362
m_surface, 362
make_workspace, 361
output_size, 361
OutputSpace, 360
surface_map, 362
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace, 1024
extension_workspace, 1025
parameter_pullback, 1025
surface_direction, 1025
surface_displacement, 1025
surface_dual, 1025
surface_pullback, 1025
surface_workspace, 1026
serif::geometry::deformation::DeformationFailure, 402
element, 403
message, 403
point, 403
reason, 403
serif::geometry::deformation::DeformationLinearization< Map >, 403
map, 404
parameters, 404
serif::geometry::deformation::DeformationMap, 245
serif::geometry::deformation::DeformationWorkspaceType< Map, Linear >, 406
Type, 407
serif::geometry::deformation::DeformationWorkspaceType< Map, true >, 407
Type, 407
serif::geometry::deformation::detail, 134
check_independent_harmonics, 135
determinant_polynomial, 135
DeterminantPolynomial, 135
evaluate_polynomial, 135
first_boundary_parameter, 136
JacobianEntries, 135
make_harmonic_modes, 136
sample_harmonic_basis, 136
validate_disjoint_output, 136
serif::geometry::deformation::DifferentiableDeformationMap, 246
serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >, 457
operator==, 457
stellar_interior, 458
surface, 458
vacuum, 458
serif::geometry::deformation::EmptyDeformationWorkspace, 492
serif::geometry::deformation::FixedInfinityRadialExtension, 537
operator==, 537
serif::geometry::deformation::GeometryStepLimit, 592
boundary_step, 593
limited_by_geometry, 593
limiting_element, 593
limiting_point, 593
limiting_point_determinant_at_step, 593
limiting_rule, 593
minimum_determinant_at_accepted, 593
minimum_determinant_at_maximum, 593
sampled_point_count, 593
step, 593
serif::geometry::deformation::GeometryStepOptions, 594
determinant_floor, 594
fraction_to_boundary, 594
maximum_step, 594
serif::geometry::deformation::keys, 137
serif::geometry::deformation::keys::DomainMap, 467
serif::geometry::deformation::keys::InteriorModel, 638
serif::geometry::deformation::keys::Linearization, 671
serif::geometry::deformation::keys::RadialInterpolation, 826
serif::geometry::deformation::keys::ReferenceDiscretization, 832
serif::geometry::deformation::keys::ReferenceSurface, 838
serif::geometry::deformation::keys::ShapeCoefficients, 914
serif::geometry::deformation::keys::SurfaceDisplacement, 938
serif::geometry::deformation::keys::SurfaceMap, 946
serif::geometry::deformation::keys::SurfaceMetric, 946
serif::geometry::deformation::keys::SurfaceModel, 947
serif::geometry::deformation::keys::VacuumModel, 980
serif::geometry::deformation::keys::VolumeDisplacement, 1014
serif::geometry::deformation::keys::VolumeExtension, 1014
serif::geometry::deformation::LinearDeformationMap, 246
serif::geometry::deformation::NodalRadialSurface, 716
operator==, 717
reference_center, 717

serif::geometry::deformation::PowerLawRadialExtension, 777
operator==, 777
radial_power, 778
serif::geometry::deformation::RadialExtension, 822
apply, 824
apply_transpose, 824
input_size, 824
InputSpace, 823
interpolation, 824
is_linear, 825
m_interpolation, 825
m_radial_weights, 825
make_radial_extension, 825
output_size, 824
OutputSpace, 823
radial_weights, 825
RadialExtension, 824
serif::geometry::deformation::RadialInterpolation, 827
infinity_radius, 828
interpolation_weights, 828
logical_radii, 828
row_offsets, 828
scalar_dof_count, 828
stellar_radius, 828
stellar_support, 828
surface_columns, 828
surface_node_count, 828
surface_scalar_dofs, 829
vacuum_support, 829
serif::geometry::deformation::ReferenceSurface, 839
positions, 839
scalar_dofs, 839
serif::geometry::deformation::spaces, 137
serif::geometry::deformation::spaces::ShapeParameters, 914
serif::geometry::deformation::spaces::SurfaceDisplacement, 938
serif::geometry::deformation::spaces::VolumeDisplacement, 1014
serif::geometry::deformation::SphericalHarmonicMode, 932
degree, 933
operator==, 933
order, 933
serif::geometry::deformation::SphericalHarmonicSurface, 933
maximum_azimuthal_order, 934
maximum_degree, 934
operator==, 934
reference_center, 934
serif::geometry::deformation::SurfaceEmbedding, 939
apply, 941
apply_transpose, 941
build_surface_embedding, 943
has_nodal_parameters, 941
input_size, 941
InputSpace, 941
is_linear, 943
modes, 942
modes_, 943
nodal_basis, 942
nodal_basis_, 943
output_size, 942
OutputSpace, 941
radial_directions, 942
radial_directions_, 943
reference_positions, 942
reference_positions_, 943
reference_radii, 942
reference_radii_, 944
surface_node_count, 942
SurfaceEmbedding, 941
serif::geometry::surface, 137
boundary_values, 139
empty_surface, 139
evaluate_pressure_surface, 139
inconsistent_sample_count, 139
Isobaric, 139
make_pressure_surface_graph, 140
non_finite_eos_response, 139
non_finite_linearization, 139
non_finite_state, 139
pressure_condition, 141
pressure_relation_t, 139
SurfaceFailureReason, 139
SurfaceLinearization, 139
SurfacePoint, 139
serif::geometry::surface::BoundaryValues< Term >, 327
operator==, 328
values, 328
serif::geometry::surface::detail, 141
accumulate_partial, 144
BoundTerms, 143
checked_linearization, 144
evaluate_relation, 144
evaluate_response_samples, 144
evaluate_surface_response, 144
finite_state, 145
is_pressure_surface_formulation, 147
is_pressure_surface_formulation< DirectPressureSurfaceFormul
EOS, Bindings, RowQuantity, Relation >
>, 147
is_pressure_surface_formulation< PressureSurfaceFormulation<
EOS, Bindings, Carrier > >, 147
linearize_carrier_samples, 145

linearize_pressure_samples, 145
linearize_surface, 145
make_surface_eos_computation, 146
make_surface_linearization_computation, 146
matches_pressure_relation, 146
relation_input, 146
state_value, 146
supports_partials, 147
surface_point, 147
surface_sample_count, 147
serif::geometry::surface::detail::AreSurfaceBindings< Bindings >, 275
serif::geometry::surface::detail::AreSurfaceBindings< utils::dependencies::Bindings< Entries... > >, 276
serif::geometry::surface::detail::BoundaryValueTracking, 329
identity, 330
serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, Catalog >, 695
serif::geometry::surface::detail::MatchingPressureRelations< EOS, Bindings, Carrier, eos::relations::RelationCatalog< Relations... > >, 695
count, 696
Matches, 696
serif::geometry::surface::detail::PressureConditionTracking, 786
capture, 786
identity, 786
matches, 786
serif::geometry::surface::detail::PressureSurfaceDescriptions< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 793
CarrierQuantity, 794
CarrierTerm, 794
EOSStateQuantities, 794
EOSStateTerms, 794
EquationOfState, 794
Operands, 794
PhysicalCondition, 795
PhysicalQuantity, 795
Relation, 795
ResidualQuantity, 795
resolves_carrier, 796
StateBindings, 795
StateQuantities, 795
StateTerms, 795
Support, 796
Test, 796
serif::geometry::surface::detail::RelationQuantities< eos::relations::Relation< Output, Inputs... > >, 845
AllInputs, 845
NonPressureInputs, 845
serif::geometry::surface::detail::RelationQuantities< Relation >, 845
serif::geometry::surface::detail::SurfaceOperands< Terms >, 947
serif::geometry::surface::detail::SurfaceOperands< utils::dependencies::KeyList< Terms... > >, 947
Type, 947
serif::geometry::surface::detail::UniqueTerms< Terms >, 978
serif::geometry::surface::detail::UniqueTerms< utils::types::TypeList< Terms... > >, 978
Type, 979
serif::geometry::surface::DirectPressureSurfaceCompilable, 247
serif::geometry::surface::DirectPressureSurfaceFormulation< EOS, Bindings, RowQuantity, Relation >, 419
serif::geometry::surface::IsobaricSurface, 645
continuation_identity, 647
descriptor, 647
IsobaricSurface, 647
m_targetPressure, 647
PhysicalQuantity, 646
target_pressure, 647
TargetValue, 646
serif::geometry::surface::IsobaricSurface::Parameters, 749
PSurf, 749
serif::geometry::surface::IsobaricSurfaceType, 247
serif::geometry::surface::keys, 148
serif::geometry::surface::keys::EOSResponse, 505
serif::geometry::surface::keys::EquationOfState, 512
serif::geometry::surface::keys::Linearization, 672
serif::geometry::surface::keys::PressureCondition, 785
serif::geometry::surface::PressureSurfaceCompilable, 247
serif::geometry::surface::PressureSurfaceDescriptor, 796
target_pressure, 797
serif::geometry::surface::PressureSurfaceFormulation< EOS, Bindings, Carrier >, 797
serif::geometry::surface::PressureSurfaceFormulationType, 248
serif::geometry::surface::SurfaceEOSResponse< Formulation >, 944
field_derivatives, 945
value, 945
serif::geometry::surface::SurfaceFailure, 945
message, 946
reason, 946
sample, 946
serif::geometry::surface::SurfaceResidualRow, 248
serif::geometry::surface::SurfaceScalarTerm, 248

serif::geometry::surface::SurfaceStateBindings, 249
serif::mapping, 148
AT_COMPACTIFIED_INFINITY, 150
compute_hdiv_mass_tensor, 150
compute_hdiv_mass_tensor_variation, 150
compute_scalar_diffusion_tensor, 150
ELEMENT_1, 150
ELEMENT_2, 150
element_displacement_data_from_element_vdofs, 151
FaceElementSide, 150
INVALID_DIMENSION, 150
INVALID_REFERENCE_RADIUS, 150
make_MappingStatus_SERiFErrorCode_map, 151
make_SERIFErrorCode_MappingStatus_map, 151
map_hcurl_curl_to_physical, 151
map_hcurl_field_to_physical, 151
map_hdiv_divergence_to_physical, 151
map_hdiv_flux_to_physical, 151
map_physical_curl_to_hcurl_reference, 152
map_physical_field_to_hcurl_reference, 152
map_physical_flux_to_hdiv_reference, 152
map_physical_gradient_to_reference, 152
map_physical_vector_gradient_to_reference, 152
map_reference_gradient_to_physical, 152
map_reference_vector_gradient_to_physical, 153
MappingStatus, 150
MappingStatus_enum_string_map, 153
MappingStatus_SERiFErrorCode_map, 153
MappingStatus_string_enum_map, 153
NON_FINITE_MAPPING_INPUT, 150
NON_FINITE_MAPPING_RESULT, 150
NON_POSITIVE_DETERMINATE, 150
OUTSIDE_REFERENCE_DOMAIN, 150
SERiFErrorCode_MappingStatus_map, 153
VALID, 150
serif::mapping::compactification, 153
serif::mapping::compactification::ExteriorDomainMap, 524
~ExteriorDomainMap, 525
clone, 525
evaluate, 525
evaluate_variation, 525
get_name, 525
is_affine_in_displacement, 525
serif::mapping::compactification::ExteriorMapDirection, 526
displaced_position_variation, 526
displacement_jacobian_variation, 526
serif::mapping::compactification::ExteriorMapInput, 526
compactification_coordinate, 527
compactification_coordinate_gradient, 527
displaced_position, 527
displacement_jacobian, 527
reference_position, 527
serif::mapping::compactification::ExteriorMapResult, 527
mapping_jacobian, 528
physical_position, 528
serif::mapping::compactification::ExteriorMapVariation, 528
mapping_jacobian_variation, 528
physical_position_variation, 528
serif::mapping::compactification::KelvinCompactification, 661
clone, 663
compute_radial_factors, 663
evaluate, 663
evaluate_jacobian_variation_factors, 663
evaluate_variation, 663
get_coordinate_tolerance, 663
get_name, 663
get_reference_infinity_radius, 664
get_reference_stellar_radius, 664
is_affine_in_displacement, 664
KelvinCompactification, 662
m_name, 664
m_options, 664
serif::mapping::compactification::KelvinCompactification::RadialFacto 826
compactification_coordinate, 826
computational_radius, 826
scale, 826
scale_derivative, 826
serif::mapping::CompactificationPointData, 356
compactification_coordinate, 356
coordinate_gradient, 356
serif::mapping::DMFixedMeshGeometry, 431
inverse_jacobian, 431
jacobian, 431
position, 431
weight, 431
serif::mapping::DMGeometryCache, 431
enabled, 432
hits, 432
mesh, 432
misses, 432
nodesSequence, 432
points, 432
sequence, 432
serif::mapping::DMGeometryCacheStatistics, 433
hits, 433

misses, 433
points, 433
serif::mapping::DMReferenceDisplacementSample, 433
gradient, 433
value, 433
serif::mapping::DMWorkspace, 434
DMWorkspace, 435
get_dimension, 435
m_compactification_dshape, 435
m_compactification_point, 435
m_compactification_shape, 435
m_dimension, 435
m_exterior_result, 435
m_exterior_variation, 436
m_field_jacobian, 436
m_field_value, 436
m_full_element_jacobian, 436
m_mapped_normal, 436
m_matrix_temp_1, 436
m_matrix_temp_2, 436
m_mesh_dshape, 436
m_reference_dshape, 436
m_reference_field_jacobian, 436
m_reference_normal, 437
m_shape, 437
m_vector_temp, 437
set_dimension, 435
serif::mapping::DomainMapper, 467
~DomainMapper, 469
clear_geometry_cache, 469
clone, 469
DomainMapper, 469
evaluate_compactification_coordinate, 470
evaluate_displacement_field, 470
evaluate_face, 470
evaluate_face_variation, 470
evaluate_point, 470
evaluate_point_impl, 471
evaluate_point_variation, 471
evaluate_point_variation_impl, 471
evaluate_volume, 471
evaluate_volume_variation, 472
get_dimension, 472
get_exterior_map, 472
get_fixed_mesh_geometry, 472
get_geometry_cache_statistics, 472
is_compactified_element, 472
is_geometry_cache_enabled, 473
m_exterior_map, 474
m_geometryCache, 474
m_options, 474
operator=, 473
select_face_element_integration_point, 473
select_face_element_transformation, 473
set_geometry_cache_enabled, 473
validate_element_data, 473
serif::mapping::ElementCompactificationData, 485
ElementCompactificationData, 485
get_dof_count, 485
get_dofs, 485
get_element, 486
m_dofs, 486
m_element, 486
serif::mapping::ElementDisplacementData, 488
ElementDisplacementData, 489
get_dimension, 489
get_dof_count, 489
get_dof_matrix, 489
get_element, 489
get_ordering, 489
m_dimension, 489
m_dof_matrix, 489
m_element, 489
m_ordering, 490
serif::mapping::ElementMappingData, 491
compactification, 492
displacement, 492
serif::mapping::FaceMappingContext, 529
mapping, 530
physical_surface_weight, 530
quadrature, 530
reference_normal, 530
reference_surface_weight, 530
serif::mapping::FaceMappingVariation, 531
mapping, 531
normal_flux_scale_variation, 531
physical_normal_variation, 531
physical_surface_weight_variation, 531
serif::mapping::FaceQuadratureContext, 532
ds, 532
normal, 532
v_dot_n_scale, 532
serif::mapping::GridFunctionMappingEvaluator, 615
evaluate_face, 616
evaluate_point, 616
evaluate_volume, 617
get_face_quadrature_context, 617
get_physical_point, 617
get_quadrature_context, 617
GridFunctionMappingEvaluator, 616
invalidate_cache, 617
invalidate_for_changed_spaces, 617
load_element, 617
m_cached_element_id, 618
m_compactification_coordinate, 618
m_compactification_data, 618
m_compactification_dofs, 618

m_compactification_space, 618
m_compactification_space_sequence, 618
m_displacement, 618
m_displacement_data, 619
m_displacement_dofs, 619
m_displacement_space, 619
m_displacement_space_sequence, 619
m_element_compactification, 619
m_element_displacement, 619
m_mapper, 619
m_workspace, 619
refresh, 618
validate_field_bindings, 618
serif::mapping::MappingPointContext, 676
compactified, 677
displaced_position, 677
displacement_jacobian, 677
inverse_mapping_jacobian, 677
mapping_determinate, 677
mapping_jacobian, 677
physical_position, 677
reference_position, 677
serif::mapping::MappingPointVariation, 678
displacement_jacobian_variation, 678
displacement_variation, 678
inverse_mapping_jacobian_variation, 678
mapping_determinate_variation, 678
mapping_jacobian_variation, 678
physical_position_variation, 679
serif::mapping::PhysicalPositionFunctionCoefficient, 756
Eval, 757
Func, 757
m_f, 757
m_mapping, 757
PhysicalPositionFunctionCoefficient, 757
serif::mapping::VolumeMappingCache, 1015
get_dimension, 1015
get_point_count, 1015
get_point_data, 1016
load, 1016
load_inverse_jacobian, 1016
m_data, 1016
m_dimension, 1016
m_point_count, 1017
m_point_stride, 1017
set_size, 1016
store, 1016
validate_point, 1016
serif::mapping::VolumeMappingContext, 1017
mapping, 1018
quadrature, 1018
serif::mapping::VolumeMappingVariation, 1018
inverse_element_jacobian_variation, 1019
mapping, 1019
weight_variation, 1019
serif::mapping::VolumeQuadratureContext, 1022
J_element_to_physical_inverse, 1022
mapping_determinate, 1022
physical_integration_weight, 1022
serif::meta, 154
enum_to_enum_string_map, 154
enum_to_string, 154
enum_to_string_enum_map, 154
template_name, 154
type_name, 154
serif::operators, 155
bind_contributions, 159
CentralDensity, 156
Closure, 156
Constant, 156
Density, 157
DensityValue, 157
Displacement, 157
DisplacementDivergence, 157
DisplacementValue, 157
Enthalpy, 157
EnthalpyGradient, 157
EnthalpyValue, 157
for_each_contribution, 159
GravityDivergence, 157
GravityValue, 157
Mass, 158
Mechanical, 158
MechanicalDifference, 158
MixedSamples, 158
Momentum, 158
Omega, 158
Potential, 158
Pressure, 158
ScaledMechanical, 158
ScaledPoisson, 158
Schema, 159
SurfacePressure, 159
serif::operators::backend, 159
serif::operators::backend::mfem, 159
accumulate_element_result, 173
add_jacobian_transpose_rows, 173
add_point_border_residual, 173
apply_jacobian, 173, 174
apply_jacobian_rows, 174
apply_jacobian_transpose, 174
apply_point_border_jacobian, 175
apply_point_border_jacobian_transpose, 175
apply_residual_rows, 175
apply_volume_direction, 175
bind, 176
Centering, 169

Closure, 169
collect_volume_results, 176
connect_volume_contributions, 177
ContractionMode, 172
Density, 170
DensityBinding, 170
DisplacementBinding, 170
element_equation_point, 177
element_jacobian_action, 177
element_jacobian_transpose, 177
element_residual, 178
element_values, 178
ElementBatch, 170
EnthalpyBinding, 170
EQUATION, 173
equation_rule, 178
evaluate_contribution_equation, 178
evaluate_element_equation, 179
evaluate_nodal_values, 179
for_each_volume_element, 179
geometry_integrand_variation, 179
index, 191
integration_order, 180
make_boundary_discretization, 180
make_contribution_computations, 180
make_nodal_dependency_context, 180
make_nodal_discretization, 180
make_nodal_equation_cache, 181
make_nodal_equation_computation, 181
make_nodal_equation_point, 181
make_operand_computations, 181
make_point_border_cache, 181
make_point_border_dependency_context, 182
make_point_border_discretization, 182
make_point_discretization, 182
make_term_coefficients, 182
make_volume_equation_cache, 182
make_volume_equation_computations, 183
make_volume_inputs, 183, 184
map_contribution_geometry, 184
map_contribution_operand, 184
map_element_geometry, 185
map_element_operand, 185
MAPPING, 173
mask_residual_dual, 185
nodal_work_report, 185
NodalCache, 170
NodalCoefficients, 170
NodalMesh, 170
NON_FINITE, 173
PreparationMode, 171
PreparationRejectionReason, 172
prepare_nodal, 185
prepare_point_border, 186
prepare_volume, 186
ReferenceOperation, 173
report_contribution_preparation, 186
sample_contribution_basis, 186
sample_contribution_operand, 187
sample_element_basis, 187
SamplesFor, 171
Schema, 171
select_nodal_values, 187
store_element_linearization, 187
store_nodal_linearization, 188
tensor_element_count, 188
try_prepare_nodal, 188
try_prepare_point_border, 188
try_prepare_validated, 189
try_prepare_volume, 189
validate_nodal_discretization, 189
validate_nodal_output, 189
validate_nodal_selection, 189
validate_point_border_outputs, 190
validate_volume_data, 190
validate_volume_discretization, 190
values, 191
Vector, 171
volume_height, 190
volume_residual, 190
volume_width, 191
volume_work_report, 191
VolumeCache, 171
VolumeCoefficients, 171
VolumeDiscretization, 191
VolumeEquationOf, 171
VolumeInputs, 172
VolumeMesh, 172
VolumeOperandResult, 172
serif::operators::backend::mfem::AffinePointEquation, 249
serif::operators::backend::mfem::BasisBindings< ColumnCount >, 290
columns, 291
compactification_space, 291
displacement_binding, 291
mapper, 291
nodes_sequence, 291
operator==, 291
residual_binding, 291
serif::operators::backend::mfem::BasisEvaluation, 300
evaluate, 303
height, 303
integrate, 303
m_components, 306
m_dense, 306
m_mapping, 306
m_points, 306

m_tensor, 306
m_width, 306
map_point, 304
reset_from, 304
sample, 304
set_constant, 305
uses_tensor, 305
width, 305
serif::operators::backend::mfem::BasisWorkspace, 308
reference_values, 308
tensor, 308
serif::operators::backend::mfem::ClosureInputsAccept, 250
serif::operators::backend::mfem::ClosureResult, 250
serif::operators::backend::mfem::CoefficientTracking, 350
capture, 351, 352
identity, 352
matches, 352
serif::operators::backend::mfem::CoefficientView< BindingT >, 353
binding, 354
Term, 354
values, 354
serif::operators::backend::mfem::CompactificationCoefficients, 356
serif::operators::backend::mfem::ContributionData< OperandCount >, 391
elements, 391
residual_offset, 391
serif::operators::backend::mfem::DisplacementCoefficients, 425
serif::operators::backend::mfem::ElementBasis< OperandCount >, 484
displacement_basis, 484
element, 484
reference_operand_basis, 484
reference_test_basis, 484
rule, 485
serif::operators::backend::mfem::ElementData< OperandCount >, 486
basis, 487
element, 487
geometry, 487
reference_operand_values, 487
validation_integral, 487
weighted_field_derivatives, 488
weighted_position_derivative, 488
weighted_residual, 488
serif::operators::backend::mfem::ElementGeometry< OperandCount >, 490
compactification, 491
displacement, 491
mapping, 491
operand_basis, 491
test_basis, 491
serif::operators::backend::mfem::EvaluableOperand, 250
serif::operators::backend::mfem::JacobianOperator< CacheT >, 657
checked_data, 659
JacobianOperator, 659
m_cache, 660
m_workspace, 660
Mult, 659
MultTranspose, 660
serif::operators::backend::mfem::MappingCoefficients, 676
binding, 676
values, 676
serif::operators::backend::mfem::MeshAndCoefficientBindings, 698
serif::operators::backend::mfem::MetadataTracking, 702
identity, 703
serif::operators::backend::mfem::nodal_keys, 192
serif::operators::backend::mfem::nodal_keys::BorderLinearization, 314
serif::operators::backend::mfem::nodal_keys::Linearization, 672
serif::operators::backend::mfem::NodalData< EquationT, SchemaT >, 713
derivative, 714
Discretization, 714
discretization, 714
residual, 714
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >, 714
components, 716
Equation, 715
residual_binding, 716
ResidualBinding, 715
selected_rows, 716
Test, 715
Value, 715
serif::operators::backend::mfem::NodalValueTracking, 720
identity, 721
serif::operators::backend::mfem::NumericalWork, 723
did_work, 724
geometry_points, 724
physics_points, 724
refreshed_scalars, 724
reused_elements, 724
sampled_operands, 724
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 728
components, 732

divergence, 732
evaluate, 730
geometry_dependent, 732
geometry_variation, 730
global_scalar, 732
gradient_trace, 730
h1_divergence, 732
hdiv, 732
integrate, 730
map_basis_point, 731
physical_value, 731
reference_components, 733
reference_value, 731
ReferenceValue, 729
sample_basis, 731
Term, 729
Value, 730
vector_copies, 733
serif::operators::backend::mfem::OperandSamples, 735
physical, 736
reference, 736
serif::operators::backend::mfem::PointBorderData< EquationT, SchemaT >, 759
Border, 760
border_binding, 760
border_coefficient, 760
nodal, 760
serif::operators::backend::mfem::PointBorderDiscretization<Equation, EquationT, SchemaT >, 760
border_binding, 761
BorderBinding, 761
Equation, 761
nodal, 761
serif::operators::backend::mfem::PreparationRejection, 780
element, 781
mapping_status, 781
message, 781
quadrature_point, 781
reason, 781
serif::operators::backend::mfem::SelectedNodalValues, 910
selected_rows, 911
values, 911
serif::operators::backend::mfem::TensorBasisCache, 957
clear, 958
get, 958
m_entries, 959
size, 959
serif::operators::backend::mfem::TensorBasisCache::Entry, 501
basis, 501
copies, 501
element, 501
operation, 501
points, 502
serif::operators::backend::mfem::volume_keys, 192
serif::operators::backend::mfem::volume_keys::Basis< Contribution >, 290
serif::operators::backend::mfem::volume_keys::Geometry< Contribution >, 591
serif::operators::backend::mfem::volume_keys::Linearization, 672
serif::operators::backend::mfem::volume_keys::PhysicalValues< Contribution, OperandT >, 759
serif::operators::backend::mfem::volume_keys::QuadratureLinearizati Contribution >, 804
serif::operators::backend::mfem::volume_keys::ReferenceValues< Contribution, OperandT >, 840
serif::operators::backend::mfem::VolumeConfiguration< EquationT, SchemaT >, 1007
discretization, 1008
DisplacementBinding, 1007
equation, 1008
options, 1008
serif::operators::backend::mfem::VolumeData< EquationT, SchemaT >, 1010
bindings, 1011
configuration, 1011
contributions, 1011
Equation, 1011
Plan, 1011
residual, 1011
Schema, 1011
serif::operators::backend::mfem::VolumeDiscretization< TestT, SchemaT >, 1012
compactification, 1013
compactification_sequence, 1013
compactification_space, 1013
displacement_binding, 1013
DisplacementBinding, 1013
mapper, 1013
residual_binding, 1013
RowBinding, 1013
serif::operators::backend::mfem::VolumeOptions, 1021
contraction_mode, 1022
quadrature_policy, 1022
serif::operators::backend::mfem::VolumeScratch, 1023
basis, 1023
workspace, 1023
serif::operators::backend::mfem::Workspace, 1026
basis, 1027
displacement_direction, 1027
displacement_dual, 1027

displacement_gradient, 1027
displacement_samples, 1027
displacement_value, 1027
element_result, 1027
element_values, 1028
full_result, 1028
mapping_context, 1028
mapping_variation, 1028
mapping_workspace, 1028
operand_dual, 1028
operand_values, 1028
quadrature_values, 1028
reference_test_values, 1028
test_basis_variation, 1028
test_values, 1029
test_variation_result, 1029
serif::operators::BadContributionCollection, 277
contributions, 278
name, 278
Test, 278
serif::operators::BadContributionRow, 279
contributions, 280
name, 280
Test, 279
serif::operators::BarotropicHydrostaticEquilibrium, 287
additional_order, 289
AngularVelocity, 288
BarotropicHydrostaticEquilibrium, 289
Constant, 288
evaluate, 289
m_rotation, 289
name, 289
Operands, 288
Potential, 288
Test, 289
serif::operators::CanSubtract, 251
serif::operators::CanSum, 251
serif::operators::ContributionBinding< EquationT >, 388
coefficient, 389
equation, 389
serif::operators::detail, 192
bind_contributions, 193
visit_contributions, 193
serif::operators::detail::AsKeyList< ListT >, 276
serif::operators::detail::AsKeyList< utils::types::TypeList< T... > >, 277
type, 277
serif::operators::detail::Columns< ListT >, 355
serif::operators::detail::Columns< utils::types::TypeList< O... > >, 355
type, 355
serif::operators::detail::ContributionChildren< EquationT >, 389
ContributionChildren, 390
parent, 390
values, 390
serif::operators::detail::ContributionOperands< std::tuple< C... > >, 392
serif::operators::detail::ContributionOperands< TupleT >, 391
serif::operators::detail::CouplingForm< Input, TestOperand >, 398
Test, 399
Trial, 399
type, 399
serif::operators::detail::Join< Lists >, 660
type, 661
serif::operators::EquationPlan< EquationT, OperandsT >, 512
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 512
column_count, 515
column_index, 514
Columns, 513
component_count, 515
component_offset, 514
for_each_column, 514
for_each_input, 514
for_each_operand, 514
Form, 513
operand_count, 515
Operands, 513
Support, 513
Test, 514
serif::operators::EquationRegistry< EquationTs >, 518
EquationRegistry, 519
Equations, 518
for_each, 519
get, 519
m_equations, 519
Rows, 518
serif::operators::GradientRequest, 601
additional_order, 602
evaluate, 602
name, 602
Operands, 601
Test, 601
serif::operators::HasScalarVolumeOperands< EquationT, OperandsT >, 623
serif::operators::HasScalarVolumeOperands< EquationT, utils::types::TypeList< Operands... > >, 624
serif::operators::NestedBadCollection, 712
contributions, 713

name, 713
Test, 712
serif::operators::NodalRequest< TermT, OperationT, PositionIndependent >, 718
evaluate, 719
name, 719
Operands, 719
position_independent, 719
Result, 719
serif::operators::NodalValueEquation, 251
serif::operators::OperandValues< OperandsT, OutputT >, 736
serif::operators::ScalarValueOperand, 252
serif::operators::ScalarValueTerm, 252
serif::operators::ScalarVolumeEquation, 252
serif::operators::VolumeContributions< std::tuple< C... > >, 1009
serif::operators::VolumeContributions< TupleT >, 1008
serif::operators::VolumeEquation, 253
serif::operators::VolumeOperand, 253
serif::operators::VolumeOperands< C, ListT >, 1019
serif::operators::VolumeOperands< C, utils::types::TypeList< O... > >, 1020
serif::utils, 193
serif::utils::checks, 193
is_finite, 194
serif::utils::CompensatedSum, 356
add, 357
m_correction, 357
m_sum, 357
value, 357
serif::utils::dependencies, 194
borrow, 199
bound_type_t, 198
CombinedKeyList, 198
computed, 199
has_binding_v, 201
InputList, 198
is_bindings_v, 201
is_bindings_v< Bindings< BindingTs... > >, 201
is_computation_graph_v, 202
is_computation_graph_v< ComputationGraph< Computations... > >, 202
is_computation_v, 202
is_computation_v< Computation< KeyT, InputsT, FunctionT > >, 202
is_computation_v< IncrementalComputation< KeyT, ValueT, InputsT, FunctionT > >, 202
is_dependency_context_v, 202
is_dependency_context_v< DependencyContext< Resources... > >, 202
is_dependency_context_v< DependencyContextView< ParentT, KeysT > >, 202
is_key_list_v, 203
is_key_list_v< KeyList< KeyTs... > >, 203
is_owned_computation_v, 203
is_owned_computation_v< OwnedComputation< KeyT, ComputationT > >, 203
is_resource_v, 203
is_resource_v< Resource< KeyT, StorageT, PolicyT > >, 203
make_computation, 199
make_computation_cache, 199
make_computation_graph, 200
make_computation_store, 200
make_dependency_context, 200
make_incremental_computation, 200
not_required, 199
own, 200
own_computation, 201
preparation_mode_name_map, 203
preparation_status_name_map, 203
PreparationMode, 198
PreparationStatus, 198
rebuild, 198
RequestedResults, 198
ResolvedTargets, 198
reuse_unchanged, 198
reused, 199
source_identity, 201
track_by, 201
serif::utils::dependencies::Binding< KeyT, TargetT >, 309
Key, 309
Target, 309
serif::utils::dependencies::Bindings< BindingTs >, 310
serif::utils::dependencies::BindingsWithBinding, 253
serif::utils::dependencies::BoundType< BindingsT, KeyT >, 330
serif::utils::dependencies::CanPrepare, 254
serif::utils::dependencies::CompatibleComputations, 254
serif::utils::dependencies::Computation< KeyT, InputsT, FunctionT >, 363
Computation, 364
evaluate, 364
Function, 363
Inputs, 363
Key, 364
m_function, 364
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 365
Bindings, 367
ComputationCache, 368

contains, 368
Failure, 367
get, 368
input, 369
inspect_source, 369
invalidate, 369
m_active, 371
m_graph, 371
m_preparation_count, 371
m_ready, 371
m_report, 372
m_storage, 372
operator=, 369
preparation_count, 370
prepare_in_order, 370
prepare_one, 370
ready, 370
Report, 367
report, 370
ResultMask, 367
source_count, 372
SourceMask, 367
Sources, 367
Storage, 367
Traits, 367
try_prepare, 371
version_index, 371
serif::utils::dependencies::ComputationGraph< Computations >, 376
computation, 377
ComputationAt, 376
ComputationGraph, 377
Dependencies, 376
index, 377
Inputs, 377
m_computations, 378
Results, 377
size, 378
serif::utils::dependencies::ComputationGraphFor, 254
serif::utils::dependencies::ComputationStore< EntryTs >, 380
Bindings, 381
ComputationStore, 381
get, 381, 382
m_entries, 382
operator=, 382
serif::utils::dependencies::ComputationUpdate< InputsT, ValueT >, 383
changed, 383
changed_inputs, 383
previous, 383
rebuild, 384
serif::utils::dependencies::DependencyContext< ResourceTs >, 413
Bindings, 413
DependencyContext, 414
get, 414
m_resources, 415
resource, 414
ResourceBindings, 413
Resources, 414
select, 414, 415
serif::utils::dependencies::DependencyContextView< ParentT, KeyList< Keys... > >, 415
Bindings, 416
DependencyContextView, 417
get, 417
m_parent, 417
resource, 417
ResourceBindings, 416
Resources, 416
serif::utils::dependencies::DependencyContextView< ParentT, KeysT >, 415
serif::utils::dependencies::detail, 204
compatible_error_v, 206
compatible_error_v< Source, std::variant< Alternatives... > >, 206
compatible_errors, 206
contains_key_v, 206
contains_key_v< KeyT, KeyList< Keys... > >, 207
independent_results, 206
key_list_subset_v, 207
key_list_subset_v< KeyList< Keys... >, AvailableT >, 207
type_index, 206
UsedResources, 205
serif::utils::dependencies::detail::AppendBinding< Bindings< BindingsTs... >, BindingT >, 270
type, 270
serif::utils::dependencies::detail::AppendBinding< BindingsT, BindingT >, 270
serif::utils::dependencies::detail::AppendKeys< ResultT, FirstT, Rest... >, 271
type, 272
serif::utils::dependencies::detail::AppendKeys< ResultT, Keys >, 271
type, 271
serif::utils::dependencies::detail::AppendUnique< KeyList< Keys... >, T >, 272
type, 273
serif::utils::dependencies::detail::AppendUnique< ListT, T >, 272
serif::utils::dependencies::detail::BorrowedValue< T >, 314

get, 315
Target, 314
value, 315
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >, 331
select, 332
target_type, 332
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< K, T >, 309
type, 309
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::collector, 354
serif::utils::dependencies::detail::BoundType< BindingsT, KeyT >, 331
serif::utils::dependencies::detail::CachedResult< ValueT, InputCount >, 332
input_versions, 333
value, 333
serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, ResourcesT, IndicesT >, 372
serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > >, 373
results, 373
sources, 373
versions, 374
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >, 374
Bindings, 375
Resources, 375
Sources, 375
valid, 375
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, Valid >, 374
valid, 374
serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, Independent >, 378
valid, 378
serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, true >, 379
valid, 379
serif::utils::dependencies::detail::Graph< Computations >, 605
acyclic, 606
edges, 607
make_order, 606
Mask, 606
Matrix, 606
Order, 606
order, 607
prerequisites, 606
required, 606
size, 607
serif::utils::dependencies::detail::HasBinding< Bindings< BindingTs... >, KeyT >, 621
value, 621
serif::utils::dependencies::detail::HasBinding< BindingsT, KeyT >, 620
serif::utils::dependencies::detail::Invocation< ComputationT, ArgumentsT, typename >, 640
valid, 640
serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< decltype(std::declval< const ComputationT & >().evaluate(std::declval< const Arguments & >()...))> >, 641
valid, 642
serif::utils::dependencies::detail::KeyListSize< KeyList< Keys... > >, 667
serif::utils::dependencies::detail::KeyListSize< KeysT >, 667
serif::utils::dependencies::detail::MergeKeyLists< ResultT, KeyList< Keys... >, Rest... >, 697
type, 698
serif::utils::dependencies::detail::MergeKeyLists< ResultT, Lists >, 696
type, 697
serif::utils::dependencies::detail::OwnedValue< T >, 748
get, 748
Target, 748
value, 748
serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, Available >, 870
valid, 870
serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, true >, 871
valid, 871
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, Done >, 872
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, false >, 872
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true >, 873
ResolvedBindings, 873
valid, 873
serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, true >, 875
valid, 876
serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, Valid >, 874

valid, 875
serif::utils::dependencies::detail::ResourceKeys< ResourcesT >, 878
serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > >, 878
trackable, 879
type, 878
serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >, 880
Error, 880
fallible, 881
valid, 881
Value, 881
serif::utils::dependencies::detail::Result< T >, 879
Error, 879
fallible, 880
valid, 880
Value, 879
serif::utils::dependencies::detail::SourceSnapshot< ResourceT >, 917
Captured, 918
captured, 918
identity, 918
serif::utils::dependencies::detail::UsedResource< GraphT >, 979
serif::utils::dependencies::detail::UsedResource< GraphT >::Predicate< ResourceT >, 778
serif::utils::dependencies::HasBinding< BindingsT, KeyT >, 620
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >, 629
evaluate, 631
evaluate_update, 631
Function, 630
IncrementalComputation, 630
Inputs, 630
Key, 630
m_function, 631
Update, 630
serif::utils::dependencies::IsBinding, 254
serif::utils::dependencies::IsBindings, 255
serif::utils::dependencies::IsComputation, 255
serif::utils::dependencies::IsComputationGraph, 255
serif::utils::dependencies::IsDependencyContext, 255
serif::utils::dependencies::IsKeyList, 256
serif::utils::dependencies::IsOwnedComputation, 256
serif::utils::dependencies::IsResource, 256
serif::utils::dependencies::KeyList< KeyTs >, 665
Keys, 666
serif::utils::dependencies::KeyListBindings< BindingsT, KeyList< KeyTs... > >, 666
serif::utils::dependencies::KeyListBindings< BindingsT, KeysT >, 666
serif::utils::dependencies::keys, 207
serif::utils::dependencies::keys::Coordinates, 397
serif::utils::dependencies::keys::Input, 633
serif::utils::dependencies::keys::Output, 744
serif::utils::dependencies::keys::Parameters, 749
serif::utils::dependencies::keys::Scale, 906
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >, 744
checked_value, 746
get, 746, 747
Key, 745
m_value, 747
operator=, 747
OwnedComputation, 746
Target, 745
serif::utils::dependencies::PreparationFailure< ErrorT >, 779
computation_index, 779
error, 779
serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >, 781
changed_sources, 783
computed_count, 782
reused_count, 782
source_changed, 782
status, 782
statuses, 783
serif::utils::dependencies::ProjectedSnapshot< FunctionT >, 800
capture, 801
m_function, 801
matches, 801
ProjectedSnapshot, 800
serif::utils::dependencies::ResolveKeys< BindingsT, KeyList< KeyTs... > >, 874
type, 874
serif::utils::dependencies::ResolveKeys< BindingsT, KeysT >, 874
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >, 876
get, 877
Key, 876
m_policy, 877
m_storage, 877
Policy, 876
policy, 877
Resource, 877
Target, 877
serif::utils::dependencies::SatisfiesKeyList, 256
serif::utils::dependencies::SnapshotPolicyFor, 256
serif::utils::dependencies::ValueSnapshot, 998
capture, 999
matches, 999
serif::utils::errors, 207
AT_COMPACTIFIED_INFINITY, 209

CACHE_FE_BINDING_ERROR, 210
COMPACTIFICATION_MISMATCH, 209
ELEMENT_NULLPTR_ERROR, 210
EOS_ERROR, 209
EOS_INVALID_THERMODYNAMIC_INPUT, 209
EOS_NONFINITE_INPUT, 209
EOS_NONFINITE_RESULT, 209
EOS_OUTSIDE_DOMAIN, 209
EOS_UNSUPPORTED_DERIVATIVE, 209
EOS_UNSUPPORTED_RELATION, 209
EOS_WRONG_INPUT_COUNT, 209
EOS_WRONG_INPUT_QUANTITY, 209
EVALUATOR_ERROR, 209
FAILED_TYPE_INVARIANT, 210
IMPROPERLY_ORDERED_REFERENCE_RADII, 209
INVALID_ARGUMENT, 209
INVALID_COMPACTIFICATION_DERIVATIVE_TYPE, 209
INVALID_COMPACTIFICATION_RANK, 209
INVALID_COMPACTIFICATION_TYPE, 209
INVALID_COORDINATE_TOLERANCE, 209
INVALID_DIMENSION, 209
INVALID_DOF_COUNT, 209
INVALID_DOF_ORDERING, 209
INVALID_DOMAIN_ERROR, 209
INVALID_ELEMENT_GEOMETRY, 209
INVALID_ELEMENT_ID_ERROR, 210
INVALID_MAPPING, 209
INVALID_QUADRATURE_ORDER, 209
INVALID_REFERENCE_RADIUS, 209
INVALID_VACUUM_ELEMENT_ATTRIBUTE_ID, 209
INVERTED_JACOBIAN, 209
JACOBIAN_ERROR, 209
JACOBIAN_SHAPE_ERROR, 209
MATRIX_SHAPE_ERROR, 210
MISC_MAPPING_ERROR, 209
MISMATCHED_MESHING_MAPS, 209
MISSING_DISPLACEMENT_FINITE_ELEMENT_SPACE, 209
MISSING_EXTERIOR_MAP, 209
MISSING_FACE_ELEMENT, 209
NEGATIVE_VALUE_ERROR, 209
NON_FINITE_DETERMINATE, 209
NON_FINITE_MAPPING_INPUT, 209
NON_FINITE_MAPPING_RESULT, 209
NON_FINITE_VALUE_ERROR, 209
NON_POSITIVE_DETERMINATE, 209
OUT_OF_RANGE_ERROR, 210
OUTSIDE_REFERENCE_DOMAIN, 209
serif_assert, 210
serif_ensure, 210
serif_error, 210
SERiFErrorCode, 208
SERiFErrorCodeNameToTypeMap, 211
SERiFErrorCodeTypeToNameMap, 211
UNKNOWN_ERROR, 209
serif::utils::errors::serif_error_printer< Message, Code >, 911
serif::utils::misc, 211
validate_finite, 211
validate_greater_than, 211
validate_less_than, 212
validate_nonnegativity, 212
serif::utils::misc::concepts, 212
ConcatT, 213
EnumerateTypesT, 213
FilterT, 213
PrependT, 213
serif::utils::misc::concepts::AllTypesAreUnique, 257
serif::utils::misc::concepts::Concat< Lists >, 384
serif::utils::misc::concepts::Concat< types::TypeList< LhsTs... >, types::TypeList< RhsTs... >, Rest... >, 384
Type, 384
serif::utils::misc::concepts::Concat< types::TypeList< Ts... > >, 385
Type, 385
serif::utils::misc::concepts::Concat<>, 385
Type, 385
serif::utils::misc::concepts::EnumerateTypes< TypeList, Descriptor >, 502
serif::utils::misc::concepts::EnumerateTypes< types::TypeList< TypeTs... >, Descriptor >, 502
Type, 502
serif::utils::misc::concepts::Filter< List, Predicate >, 535
serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate >, 536
FilteredTail, 536
Type, 536
serif::utils::misc::concepts::Filter< types::TypeList<>, Predicate >, 536
Type, 537
serif::utils::misc::concepts::IsComparableToNumeric, 257
serif::utils::misc::concepts::IsEnum, 257
serif::utils::misc::concepts::IsException, 257
serif::utils::misc::concepts::IsExceptionWithErrorCode, 258
serif::utils::misc::concepts::IsNumeric, 258
serif::utils::misc::concepts::IsOrderable, 258
serif::utils::misc::concepts::Prepend< T, List >, 783

serif::utils::misc::concepts::Prepend< T, types::TypeList< Ts... > >, 783
Type, 784
serif::utils::misc::concepts::SetOfOneOrTwo, 258
serif::utils::misc::format, 213
join, 214
serif::utils::misc::std_helper, 214
AIsBaseClassOfB, 215
CountVariadicArguments, 215
NeitherConstNorVolatile, 214
NeitherConstNorVolatileReference, 214
SameType, 215
serif::utils::misc::terminal, 215
a_black, 217
a_blue, 217
a_bright_black, 217
a_bright_green, 217
a_bright_red, 218
a_cyan, 218
a_green, 218
a_magenta, 218
a_red, 218
a_reset, 218
a_white, 218
a_yellow, 218
ANSIColors, 216
BLACK, 217
BLUE, 217
BRIGHT_BLACK, 217
BRIGHT_GREEN, 217
BRIGHT_RED, 217
CYAN, 217
GREEN, 217
MAGENTA, 217
RED, 217
RESET, 217
resolve_ansi_color, 217
supports_ansi, 217
WHITE, 217
YELLOW, 217
serif::utils::misc::traits, 219
all_unique_types_v, 219
all_unique_types_v< First, Rest... >, 219
serif::utils::types, 219
serif::utils::types::backend, 219
serif::utils::types::backend::mfem, 220
serif::utils::types::ct_error_string< N >, 399
ct_error_string, 400
data, 400
padding, 400
view, 400
serif::utils::types::ctstring< N >, 400
ctstring, 401
data, 401
view, 401
serif::utils::types::DenseMatrixBase< T >, 407
Type, 408
serif::utils::types::DenseTensorBase< T, Rank >, 408
rank, 408
Type, 408
serif::utils::types::SparseMatrixBase< T >, 920
Type, 920
serif::utils::types::SparseTensorBase< T, Rank >, 920
rank, 921
Type, 920
serif::utils::types::TypeList< Ts >, 978
serif::utils::types::VectorBase< T >, 1003
Type, 1003
serif::utils::user_options, 220
serif::utils::user_options::DomainMapperOptions, 474
dimension, 474
vacuum_element_attribute_ID, 474
serif::utils::user_options::KelvinCompactificationOptions, 665
coordinate_tolerance, 665
r_inf_ref, 665
r_star_ref, 665
serif::utils::warnings, 220
serif_warning, 220
serif_assert
serif::utils::errors, 210
serif_ensure
serif::utils::errors, 210
serif_error
serif::utils::errors, 210
serif_warning
serif::utils::warnings, 220
SERiFErrorCode
serif::eos::models, 108
serif::utils::errors, 208
views.cpp, 1038
SERiFErrorCode_MappingStatus_map
serif::mapping, 153
SERiFErrorCodeNameToTypeMap
serif::utils::errors, 211
SERiFErrorCodeTypeToNameMap
serif::utils::errors, 211
set_constant
serif::discretization::backend::mfem::BasisEvaluation, 298
serif::operators::backend::mfem::BasisEvaluation, 305
set_dimension
serif::mapping::DMWorkspace, 435
set_fallback_control
serif::discretization::quadrature::RuleSet, 893
set_form_control

serif::discretization::quadrature::RuleSet, 893
set_geometry_cache_enabled
serif::mapping::DomainMapper, 473
set_integration_rule
serif::discretization::quadrature::backend::mfem, 106
set_role_control
serif::discretization::quadrature::RuleSet, 893
set_size
serif::mapping::VolumeMappingCache, 1016
shape_equilibrium_term
serif::discretization::blocks::surface_deformation, 938
share_reference_result
serif::geometry::deformation::backend::mfem::detail, 134
size
serif::discretization::backend::mfem::DofMap, 443
serif::discretization::backend::mfem::DofSelection, 447
serif::discretization::backend::mfem::TensorBasisCache, 957
serif::eos::relations::RelationCatalog< Relations >, 841
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnapshot, 838
serif::operators::backend::mfem::TensorBasisCache, 959
serif::utils::dependencies::ComputationGraph< Computations >, 378
serif::utils::dependencies::detail::Graph< Computations >, 607
source_changed
serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >, 782
source_count
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 372
source_identity
serif::utils::dependencies, 201
SourceMask
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 367
Sources
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 367
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >, 375
sources
serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > >, 373
space
serif::discretization::backend::mfem::DofMap, 443
serif::discretization::quadrature::backend::mfem::DiscreteFunctio
functional_space::DiscreteFunctionalSpace< SpaceV, OrderV, RankV > >, 423
spaces
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza
833
specific_term
serif::discretization::blocks::enthalpy, 497
SpecificEnergyValue
serif::dimensions, 66
SpecificEnthalpyFromDensity
serif::eos::relations, 111
SpecificEnthalpyFromPressure
serif::eos::relations, 111
SpecificEnthalpyValue
serif::dimensions, 66
SpecificEntropyValue
serif::dimensions, 66
SpecificInternalEnergyValue
serif::dimensions, 66
src/compile_time_checks/serif/dimensions/quantities.cpp, 1031
src/compile_time_checks/serif/discretization/blocks/fields.cpp, 1032
src/compile_time_checks/serif/discretization/forms/forms.cpp, 1032
src/compile_time_checks/serif/discretization/forms/operands.cpp, 1033
src/compile_time_checks/serif/discretization/quadrature/backend/n 1033
src/compile_time_checks/serif/discretization/quadrature/backend/n 1034
src/compile_time_checks/serif/eos/evaluation.cpp, 1034
src/compile_time_checks/serif/eos/models/traits.cpp, 1035
src/compile_time_checks/serif/eos/runtime/details.cpp, 1036
src/compile_time_checks/serif/eos/runtime/views.cpp, 1037
src/compile_time_checks/serif/geometry/deformation.cpp, 1038
src/compile_time_checks/serif/geometry/surface.cpp, 1039
src/compile_time_checks/serif/operators/computation_owners.cpp, 1040
src/compile_time_checks/serif/operators/plan.cpp, 1041
src/compile_time_checks/serif/operators/serial_backend.cpp, 1042
src/compile_time_checks/serif/utils/dependencies/binding.cpp,

1043
src/compile_time_checks/serif/utils/dependencies/constant_integral.cpp, 1044
src/compile_time_checks/serif/utils/dependencies/constant_scalar.cpp, 1044
src/compile_time_checks/serif/utils/dependencies/dependency_constant.cpp, 1045
src/compile_time_checks/serif/utils/dependencies/keys_list.cpp, 1046
src/compile_time_checks/serif/utils/dependencies/mixed_constant.cpp, 1047
src/include/serif/dimensions/dimensions.hpp, 1048
src/include/serif/dimensions/partial.hpp, 1051
src/include/serif/dimensions/quantities.hpp, 1053
src/include/serif/dimensions/runtime/concepts.hpp, 1281
src/include/serif/dimensions/runtime/runtime.hpp, 1075
src/include/serif/dimensions/type_alias.hpp, 1054
src/include/serif/discretization/backend/mfem/basis_evaluation.hpp, 1057
src/include/serif/discretization/backend/mfem/coefficient_traits.hpp, 1059
src/include/serif/discretization/backend/mfem/dof_mapping.hpp, 1060
src/include/serif/discretization/backend/mfem/dof_selection.hpp, 1062
src/include/serif/discretization/backend/mfem/mfem_backend.hpp, 1063
src/include/serif/discretization/backend/mfem/point_dofs.hpp, 1064
src/include/serif/discretization/backend/mfem/tensor_basis.hpp, 1066
src/include/serif/discretization/backend/mfem/term_binding.hpp, 1068
src/include/serif/discretization/blocks/base.hpp, 1069
src/include/serif/discretization/blocks/blocks.hpp, 1070
src/include/serif/discretization/blocks/concepts.hpp, 1283
src/include/serif/discretization/blocks/fields.hpp, 1071
src/include/serif/discretization/blocks/runtime.hpp, 1077
src/include/serif/discretization/blocks/traits.hpp, 1083
src/include/serif/discretization/domain/concepts.hpp, 1285
src/include/serif/discretization/domain/ids/boundary.hpp, 1092
src/include/serif/discretization/domain/ids/concepts.hpp, 1286
src/include/serif/discretization/domain/ids/domain.hpp, 1095
src/include/serif/discretization/domain/ids/lists/concepts.hpp, 1288
src/include/serif/discretization/domain/ids/lists/lists.hpp, 1097
src/include/serif/discretization/domain/ids/lists/traits.hpp, 1085
src/include/serif/discretization/domain/ids/validation/concepts.hpp, 1290
src/include/serif/discretization/domain/ids/validation/unique_ids.hpp, 1099
src/include/serif/discretization/domain/mesh/topology.hpp, 1100
src/include/serif/discretization/domain/physical_domains.hpp, 1101
src/include/serif/discretization/domain/relation/concepts.hpp, 1292
src/include/serif/discretization/domain/relation/lists/concepts.hpp, 1293
src/include/serif/discretization/domain/relation/lists/relation_list.hpp, 1102
src/include/serif/discretization/domain/relation/lists/traits.hpp, 1086
src/include/serif/discretization/domain/relation/relations.hpp, 1103
src/include/serif/discretization/domain/relation/validation/concepts.hpp, 1294
src/include/serif/discretization/domain/relation/validation/runtime.hpp, 1079
src/include/serif/discretization/domain/relation/validation/validation.hpp, 1108
src/include/serif/discretization/domain/resolver.hpp, 1110
src/include/serif/discretization/domain/schema/concepts.hpp, 1295
src/include/serif/discretization/domain/schema/domain_schema.hpp, 1111
src/include/serif/discretization/domain/schema/schemas.hpp, 1113
src/include/serif/discretization/domain/schema/traits.hpp, 1088
src/include/serif/discretization/domain/schema/utils.hpp, 1115
src/include/serif/discretization/domain/schema/validation/all.hpp, 1116
src/include/serif/discretization/domain/schema/validation/boundary.hpp, 1094
src/include/serif/discretization/domain/schema/validation/connected.hpp, 1117
src/include/serif/discretization/domain/schema/validation/inscribed.hpp, 1118
src/include/serif/discretization/domain/schema/validation/results.hpp, 1120
src/include/serif/discretization/domain/schema/validation/validator.hpp,

1121
src/include/serif/discretization/domain/types.hpp, 1123
src/include/serif/discretization/forms/concept.hpp, 1124
src/include/serif/discretization/forms/form_traits.hpp, 1126
src/include/serif/discretization/forms/forms.hpp, 1128
src/include/serif/discretization/forms/operands.hpp, 1131
src/include/serif/discretization/forms/runtime.hpp, 1081
src/include/serif/discretization/forms/value.hpp, 1132
src/include/serif/discretization/functional_space/spaces.hpp, 1135
src/include/serif/discretization/quadrature/backend/mfem/concepts.hpp, 1297
src/include/serif/discretization/quadrature/backend/mfem/config.hpp, 1138
src/include/serif/discretization/quadrature/backend/mfem/mfem.hpp, 1139
src/include/serif/discretization/quadrature/backend/mfem/mfem_rules.hpp, 1141
src/include/serif/discretization/quadrature/backend/order.hpp, 1143
src/include/serif/discretization/quadrature/policy.hpp, 1145
src/include/serif/discretization/quadrature/query.hpp, 1146
src/include/serif/discretization/quadrature/rules.hpp, 1148
src/include/serif/eos/evaluation.hpp, 1212
src/include/serif/eos/exceptions.hpp, 1351
src/include/serif/eos/models/concepts.hpp, 1299
src/include/serif/eos/models/polytropic.hpp, 1150
src/include/serif/eos/models/traits.hpp, 1089
src/include/serif/eos/relations/catalog.hpp, 1151
src/include/serif/eos/relations/concepts.hpp, 1300
src/include/serif/eos/relations/relations.hpp, 1105
src/include/serif/eos/relations/traits.hpp, 1091
src/include/serif/eos/runtime/concepts.hpp, 1301
src/include/serif/eos/runtime/details.hpp, 1209
src/include/serif/eos/runtime/views.hpp, 1152
src/include/serif/equations/angular_momentum.hpp, 1154
src/include/serif/equations/arithmetic.hpp, 1155
src/include/serif/equations/barotropic_closure.hpp, 1156
src/include/serif/equations/centering.hpp, 1157
src/include/serif/equations/central_density.hpp, 1158
src/include/serif/equations/concepts.hpp, 1303
src/include/serif/equations/contributions.hpp, 1306
src/include/serif/equations/displacement_force.hpp, 1159
src/include/serif/equations/gravity_force.hpp, 1160
src/include/serif/equations/hydrostatic_equilibrium.hpp, 1161
src/include/serif/equations/mass_normalization.hpp, 1162
src/include/serif/equations/mixed_gravity.hpp, 1163
src/include/serif/equations/point.hpp, 1164
src/include/serif/equations/pressure_force.hpp, 1165
src/include/serif/equations/pressure_surface.hpp, 1166
src/include/serif/equations/registry.hpp, 1167
src/include/serif/equations/rigid_rotation.hpp, 1169
src/include/serif/equations/rotational_force.hpp, 1170
src/include/serif/equations/sum.hpp, 1171
src/include/serif/geometry/deformation/actions.hpp, 1171
src/include/serif/geometry/deformation/backend/mfem/computation
src/include/serif/geometry/deformation/backend/mfem/detail/radial
src/include/serif/geometry/deformation/backend/mfem/geometry_st
src/include/serif/geometry/deformation/backend/mfem/inputs.hpp, 1182
src/include/serif/geometry/deformation/backend/mfem/radial_inter
src/include/serif/geometry/deformation/backend/mfem/reference_ge
src/include/serif/geometry/deformation/backend/mfem/surface_met
src/include/serif/geometry/deformation/backend/mfem/volume_disp
src/include/serif/geometry/deformation/composition.hpp, 1188
src/include/serif/geometry/deformation/computations.hpp, 1175
src/include/serif/geometry/deformation/deformation.hpp, 1190
src/include/serif/geometry/deformation/detail/action_storage.hpp, 1190
src/include/serif/geometry/deformation/detail/determinant_polynor
src/include/serif/geometry/deformation/detail/harmonics.hpp, 1193
src/include/serif/geometry/deformation/failure.hpp, 1194
src/include/serif/geometry/deformation/geometry_step.hpp, 1181

src/include/serif/geometry/deformation/linearization.hpp, 1196
src/include/serif/geometry/deformation/models.hpp, 1199
src/include/serif/geometry/deformation/radial_extension.hpp, 1200
src/include/serif/geometry/deformation/reference_surface.hpp, 1201
src/include/serif/geometry/deformation/spaces.hpp, 1203
src/include/serif/geometry/deformation/surface_embedding.hpp, 1203
src/include/serif/geometry/surface/compilation.hpp, 1205
src/include/serif/geometry/surface/computations.hpp, 1176
src/include/serif/geometry/surface/detail/batches.hpp, 1207
src/include/serif/geometry/surface/detail/evaluation.hpp, 1213
src/include/serif/geometry/surface/detail/relations.hpp, 1106
src/include/serif/geometry/surface/details.hpp, 1212
src/include/serif/geometry/surface/evaluation.hpp, 1215
src/include/serif/geometry/surface/isobaric.hpp, 1216
src/include/serif/geometry/surface/linearization.hpp, 1197
src/include/serif/geometry/surface/samples.hpp, 1219
src/include/serif/geometry/surface/surface.hpp, 1221
src/include/serif/mapping/coefficients.hpp, 1222
src/include/serif/mapping/compactification/compactification.hpp, 1223
src/include/serif/mapping/compactification/input_result.hpp, 1224
src/include/serif/mapping/compactification/kelvin.hpp, 1226
src/include/serif/mapping/compactification/variation.hpp, 1228
src/include/serif/mapping/context.hpp, 1230
src/include/serif/mapping/data.hpp, 1231
src/include/serif/mapping/domain_mapper.hpp, 1232
src/include/serif/mapping/evaluator.hpp, 1234
src/include/serif/mapping/status.hpp, 1235
src/include/serif/mapping/transformations.hpp, 1237
src/include/serif/mapping/variation.hpp, 1229
src/include/serif/mapping/volume_cache.hpp, 1239
src/include/serif/meta/enums.hpp, 1240
src/include/serif/meta/names.hpp, 1242
src/include/serif/operators/backend/mfem/element_data.hpp, 1243
src/include/serif/operators/backend/mfem/element_evaluation.hpp, 1244
src/include/serif/operators/backend/mfem/element_preparation.hpp, 1246
src/include/serif/operators/backend/mfem/element_transpose.hpp, 1248
src/include/serif/operators/backend/mfem/jacobian_operator.hpp, 1250
src/include/serif/operators/backend/mfem/nodal_computations.hpp, 1251
src/include/serif/operators/backend/mfem/nodal_data.hpp, 1252
src/include/serif/operators/backend/mfem/nodal_evaluation.hpp, 1254
src/include/serif/operators/backend/mfem/nodal_inputs.hpp, 1255
src/include/serif/operators/backend/mfem/nodal_rows.hpp, 1257
src/include/serif/operators/backend/mfem/operand_evaluation.hpp, 1259
src/include/serif/operators/backend/mfem/point_border.hpp, 1261
src/include/serif/operators/backend/mfem/preparation.hpp, 1262
src/include/serif/operators/backend/mfem/quadrature.hpp, 1265
src/include/serif/operators/backend/mfem/rejection.hpp, 1266
src/include/serif/operators/backend/mfem/term_coefficients.hpp, 1268
src/include/serif/operators/backend/mfem/volume_application.hpp, 1269
src/include/serif/operators/backend/mfem/volume_computations.hpp, 1271
src/include/serif/operators/backend/mfem/volume_data.hpp, 1273
src/include/serif/operators/backend/mfem/volume_inputs.hpp, 1276
src/include/serif/operators/backend/mfem/volume_stages.hpp, 1277
src/include/serif/operators/backend/mfem/workspace.hpp, 1279
src/include/serif/operators/concepts.hpp, 1304
src/include/serif/operators/contributions.hpp, 1307
src/include/serif/operators/plan.hpp, 1309
src/include/serif/utils/checks/finite.hpp, 1355
src/include/serif/utils/dependencies/binding.hpp, 1310
src/include/serif/utils/dependencies/computation.hpp, 1312
src/include/serif/utils/dependencies/computation_cache.hpp, 1314

src/include/serif/utils/dependencies/computation_graph.hpp, 1316
src/include/serif/utils/dependencies/computation_store.hpp, 1318
src/include/serif/utils/dependencies/dependencies.hpp, 1320
src/include/serif/utils/dependencies/dependency_context.hpp, 1324
src/include/serif/utils/dependencies/detail/cache_storage.hpp, 1325
src/include/serif/utils/dependencies/detail/cache_traits.hpp, 1327
src/include/serif/utils/dependencies/detail/graph.hpp, 1329
src/include/serif/utils/dependencies/detail/result.hpp, 1330
src/include/serif/utils/dependencies/detail/schema.hpp, 1332
src/include/serif/utils/dependencies/detail/type_lists.hpp, 1334
src/include/serif/utils/dependencies/incremental_computation.hpp, 1336
src/include/serif/utils/dependencies/key_list.hpp, 1338
src/include/serif/utils/dependencies/key_list_operations.hpp, 1340
src/include/serif/utils/dependencies/keys.hpp, 1341
src/include/serif/utils/dependencies/preparation.hpp, 1264
src/include/serif/utils/dependencies/resource.hpp, 1342
src/include/serif/utils/dependencies/snapshot.hpp, 1344
src/include/serif/utils/error/codes.hpp, 1345
src/include/serif/utils/error/errors.hpp, 1347
src/include/serif/utils/format/strings.hpp, 1363
src/include/serif/utils/misc/compensated_sum.hpp, 1348
src/include/serif/utils/misc/concepts/concat.hpp, 1349
src/include/serif/utils/misc/concepts/enumeration.hpp, 1350
src/include/serif/utils/misc/concepts/exceptions.hpp, 1352
src/include/serif/utils/misc/concepts/filter.hpp, 1353
src/include/serif/utils/misc/concepts/numeric.hpp, 1354
src/include/serif/utils/misc/concepts/type_uniqueness.hpp, 1361
src/include/serif/utils/misc/finite.hpp, 1356
src/include/serif/utils/misc/std_helper/cleaning.hpp, 1357
src/include/serif/utils/misc/std_helper/variadic_cleaning.hpp, 1358
src/include/serif/utils/misc/terminal/colors.hpp, 1359
src/include/serif/utils/misc/traits/type_uniqueness.hpp, 1362
src/include/serif/utils/types/strings.hpp, 1364
src/include/serif/utils/types/tensor/backend/mfem/tensor.hpp, 1365
src/include/serif/utils/types/tensor/tensor.hpp, 1366
src/include/serif/utils/types/type_list.hpp, 1367
src/include/serif/utils/user_options/mapping/compactification/kelvi 1227
src/include/serif/utils/user_options/mapping/domain_mapper.hpp, 1233
src/include/serif/utils/warnings/warning.hpp, 1368
src/lib/serif/discretization/backend/mfem/basis_evaluation.cpp, 1370
src/lib/serif/discretization/domain/mesh/topology.cpp, 1370
src/lib/serif/discretization/domain/schema/validation/results.cpp, 1371
src/lib/serif/discretization/quadrature/policy.cpp, 1372
src/lib/serif/eos/models/polytropic.cpp, 1373
src/lib/serif/geometry/deformation/backend/mfem/geometry_step.c 1374
src/lib/serif/geometry/deformation/backend/mfem/inputs.cpp, 1375
src/lib/serif/geometry/deformation/backend/mfem/radial_interpolat 1375
src/lib/serif/geometry/deformation/backend/mfem/radial_locator.cp 1376
src/lib/serif/geometry/deformation/backend/mfem/reference_geome 1377
src/lib/serif/geometry/deformation/backend/mfem/surface_metric.c 1378
src/lib/serif/geometry/deformation/determinant_polynomial.cpp, 1379
src/lib/serif/geometry/deformation/harmonics.cpp, 1380
src/lib/serif/geometry/deformation/radial_extension.cpp, 1381
src/lib/serif/geometry/deformation/surface_embedding.cpp, 1381
src/lib/serif/geometry/surface/isobaric.cpp, 1382
src/lib/serif/mapping/coefficients.cpp, 1383
src/lib/serif/mapping/compactification/kelvin.cpp, 1383
src/lib/serif/mapping/data.cpp, 1384
src/lib/serif/mapping/domain_mapper.cpp, 1384
src/lib/serif/mapping/evaluator.cpp, 1385
src/lib/serif/mapping/transformations.cpp, 1385
src/lib/serif/mapping/volume_cache.cpp, 1387

state_value
serif::geometry::surface::detail, 146
StateBindings
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 795
StateQuantities
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 795
StateTerms
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 795
static_block_size
serif::discretization::blocks::generated_residual_block< GeneratedResidual >, 589
serif::discretization::blocks::generated_value_block< GeneratedValue >, 591
serif::discretization::blocks::global_scalar::residual, 860
serif::discretization::blocks::global_scalar::value, 989
serif::discretization::blocks::residual_block_base, 869
serif::discretization::blocks::value_block_base, 998
status
serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >, 782
statuses
serif::utils::dependencies::PreparationReport< ResultsT, SourcesT >, 783
std::formatter< serif::discretization::blocks::FieldDescriptor >, 543
format, 544
std::formatter< serif::discretization::blocks::TermDescriptor >, 544
format, 545
std::formatter< serif::discretization::domain::DomainOrSet >, 546
format, 547
std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult >, 547
format, 547
parse, 547
string_formatter, 548
std::formatter< serif::discretization::forms::FormDescriptor >, 548
format, 549
std::formatter< serif::discretization::functional_space::Continuity >, 550
format, 551
std::formatter< serif::discretization::functional_space::DiscreteFunctionSpace< Space, Order, R > >, 551
format, 552
std::formatter< serif::discretization::functional_space::Family<>, 553
format, 554
std::formatter< serif::discretization::functional_space::Mapping >, 554
format, 555
std::formatter< serif::discretization::functional_space::Rank< R > >, 556
format, 557
std::formatter< serif::discretization::functional_space::SpaceDescriptor >, 557
format, 558
std::formatter< serif::discretization::functional_space::Trace >, 559
format, 560
std::formatter< serif::discretization::quadrature::backend::mfem::MFEMRule >, 560
format, 561
std::formatter< serif::discretization::quadrature::FormRule >, 561
format, 562
std::formatter< serif::discretization::quadrature::MappingKind >, 563
format, 564
std::formatter< serif::discretization::quadrature::Mode >, 564
format, 565
std::formatter< serif::discretization::quadrature::QuadratureRole >, 565
format, 566
std::formatter< serif::discretization::quadrature::Query >, 567
format, 568
std::formatter< serif::discretization::quadrature::RuleControl >, 568
format, 569
std::formatter< serif::discretization::quadrature::RuleSet >, 569
format, 570
stellar_attributes
serif::geometry::deformation::backend::mfem::ReferenceDiscretization< 831
serif::geometry::deformation::backend::mfem::ReferenceDiscretization< 833
stellar_interior
serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >, 458
stellar_radius
serif::geometry::deformation::RadialInterpolation, 828
stellar_support
serif::geometry::deformation::RadialInterpolation,

828
StellarDomains
serif::discretization::domain, 78
step
serif::geometry::deformation::GeometryStepLimit, 593
Storage
serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >, 737
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 367
store
serif::mapping::VolumeMappingCache, 1016
store_element_linearization
serif::operators::backend::mfem, 187
store_nodal_linearization
serif::operators::backend::mfem, 188
String2ContinuityMap
serif::discretization::functional_space, 101
String2FamilyMap
serif::discretization::functional_space, 102
String2MappingMap
serif::discretization::functional_space, 102
String2TraceMap
serif::discretization::functional_space, 102
string_formatter
std::formatter< serif::discretization::domain::schema::validation::SchemaValidationResult >, 548
sum_dynamic_orders
serif::discretization::quadrature::backend::mfem, 106
Support
serif::discretization::blocks::barotropic_constant::mass_coordinate, 689
serif::discretization::blocks::central_density::border, 313
serif::discretization::blocks::density::mass, 683
serif::discretization::blocks::displacement::vector, 1002
serif::discretization::blocks::enthalpy::specific, 925
serif::discretization::blocks::gravity::gradient, 598
serif::discretization::blocks::gravity::potential, 773
serif::discretization::blocks::rotation::angular_velocity, 262
serif::discretization::blocks::SpatialTermTraits< TermT >, 922
serif::equations::Contribution< TestOperandT, SupportT >, 388
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 796
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 513
SupportOf
serif::discretization::blocks, 76
serif::equations, 117
supports
serif::discretization::backend::mfem::DofMap, 443
serif::eos::runtime::EOSView, 508
supports_ansi
serif::utils::misc::terminal, 217
supports_partial_derivative_v
serif::eos::models, 109
supports_partials
serif::geometry::surface::detail, 147
supports_relation_v
serif::eos::models, 109
surface
serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >, 458
surface_attribute
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza 831
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza 833
surface_columns
serif::geometry::deformation::RadialInterpolation, 828
surface_direction
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace, 1025
surface_displacement
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace, 1025
surface_dual
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace, 1025
surface_map
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >, 362
surface_node_count
serif::geometry::deformation::RadialInterpolation, 828
serif::geometry::deformation::SurfaceEmbedding, 942
surface_parameter_mass_diagonal
serif::geometry::deformation::backend::mfem, 131
surface_point
serif::geometry::surface::detail, 147
surface_pullback

serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace, 1025
surface_sample_count
serif::geometry::surface::detail, 147
surface_scalar_dofs
serif::geometry::deformation::RadialInterpolation, 829
surface_workspace
serif::geometry::deformation::ComposedDeformation< SurfaceMap, ExtensionMap >::Workspace, 1026
SurfaceEmbedding
serif::geometry::deformation::SurfaceEmbedding, 941
SurfaceFailureReason
serif::geometry::surface, 139
SurfaceLinearization
serif::geometry::surface, 139
SurfacePoint
serif::geometry::surface, 139
SurfacePressure
serif::operators, 159
symbol
serif::discretization::blocks::barotropic_constant::mass_conservation, 689
serif::discretization::blocks::central_density::border, 313
serif::discretization::blocks::density::mass, 683
serif::discretization::blocks::displacement::vector, 1003
serif::discretization::blocks::enthalpy::specific, 925
serif::discretization::blocks::gravity::gradient, 598
serif::discretization::blocks::gravity::potential, 773
serif::discretization::blocks::rotation::angular_velocity, 262

tangential
serif::discretization::functional_space, 99
tangential_continuous
serif::discretization::functional_space, 98
Target
serif::utils::dependencies::Binding< KeyT, TargetT >, 309
serif::utils::dependencies::detail::BorrowedValue< T >, 314
serif::utils::dependencies::detail::OwnedValue< T >, 748
serif::utils::dependencies::OwnedComputation< KeyT, ComputationT >, 745
serif::utils::dependencies::Resource< KeyT, StorageT, PolicyT >, 877
target_density
serif::equations::CentralDensityConstraint< EOST >, 341
target_enthalpy
serif::equations::CentralDensityConstraint< EOST >, 342
target_pressure
serif::equations::PressureSurfaceConstraint< EOST >, 792
serif::geometry::surface::IsobaricSurface, 647
serif::geometry::surface::PressureSurfaceDescriptor, 797
target_type
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >, 332
TargetValue
serif::geometry::surface::IsobaricSurface, 646
TemperatureValue
serif::dimensions, 66
template_name
serif::meta, 154
tensor
serif::discretization::backend::mfem::BasisWorkspace, 307
serif::operators::backend::mfem::BasisWorkspace, 308
tensor_element_count
serif::operators::backend::mfem, 188
Term
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >, 966
serif::operators::backend::mfem::CoefficientView< BindingT >, 354
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 729
term_descriptor
serif::discretization::blocks, 77
term_type
serif::discretization::forms::Operand< TermT, OperationT >, 727
TermBinding
serif::discretization::backend::mfem::TermBinding< TermT, SchemaT >, 966, 967
Terms
serif::discretization::blocks::barotropic_constant, 281
serif::discretization::blocks::central_density, 338
serif::discretization::blocks::density, 410
serif::discretization::blocks::displacement, 425
serif::discretization::blocks::enthalpy, 497
serif::discretization::blocks::gravity, 610
serif::discretization::blocks::rotation, 887

serif::discretization::blocks::surface_deformation, 937
TermsOf
serif::discretization::blocks, 76
Test
serif::equations::BarotropicClosure< EOST >, 283
serif::equations::BarotropicHydrostaticEquilibrium, 286
serif::equations::Contribution< TestOperandT, SupportT >, 388
serif::equations::DisplacementForce< EOST >, 427
serif::equations::EquationDifference< LeftT, RightT >, 511
serif::equations::EquationScale< EquationT >, 521
serif::equations::EquationSum< FirstT, RestTs >, 523
serif::equations::GravityGradientRelation, 614
serif::equations::PoissonBalance, 764
serif::geometry::surface::detail::PressureSurfaceDescription< EOS, Bindings, RowQuantity, RelationT, ResolveCarrier >, 796
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >, 715
serif::operators::BadContributionCollection, 278
serif::operators::BadContributionRow, 279
serif::operators::BarotropicHydrostaticEquilibrium, 289
serif::operators::detail::CouplingForm< Input, TestOperand >, 399
serif::operators::EquationPlan< EquationT, utils::types::TypeList< O... > >, 514
serif::operators::GradientRequest, 601
serif::operators::NestedBadCollection, 712
test
serif::discretization::forms::FormDescriptor, 572
test_basis
serif::operators::backend::mfem::ElementGeometry< OperandCount >, 491
test_basis_variation
serif::operators::backend::mfem::Workspace, 1028
test_order
serif::discretization::quadrature::Query, 821
test_order_reduction
serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > >, 575
serif::discretization::forms::FormTraits< divergence< TrialT, TestT > >, 576
serif::discretization::forms::FormTraits< force< TrialT, TestT > >, 578
serif::discretization::forms::FormTraits< FormT >, 574
serif::discretization::forms::FormTraits< mass< TrialT, TestT > >, 579
serif::discretization::forms::FormTraits< projection< TrialT, TestT > >, 581
serif::discretization::forms::FormTraits< source< TrialT, TestT > >, 582
serif::discretization::quadrature::Query, 821
test_type
serif::discretization::forms::boundary_flux< TrialT, TestT >, 317
serif::discretization::forms::divergence< TrialT, TestT >, 430
serif::discretization::forms::force< TrialT, TestT >, 541
serif::discretization::forms::mass< TrialT, TestT >, 687
serif::discretization::forms::projection< TrialT, TestT >, 803
serif::discretization::forms::source< TrialT, TestT >, 916
test_values
serif::operators::backend::mfem::Workspace, 1029
test_variation_result
serif::operators::backend::mfem::Workspace, 1029
TestOperand
serif::equations::Contribution< TestOperandT, SupportT >, 388
TestOperandOf
serif::equations, 117
ThermodynamicQuantityID
serif::dimensions::runtime::ThermodynamicQuantityID, 974
thermodynamicQuantityID
serif::dimensions::runtime, 68
TimeValue
serif::dimensions, 66
Todo List, 3
TorqueValue
serif::dimensions, 67
Trace
serif::discretization::functional_space, 99
trace
serif::discretization::functional_space::SpaceDescriptor, 919
TraceStringMap
serif::discretization::functional_space, 102
track_by
serif::utils::dependencies, 201
trackable
serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > >, 879

tracks_coordinates
serif::discretization::backend::mfem::DofSelection, 448
Traits
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 367
Trial
serif::operators::detail::CouplingForm< Input, TestOperand >, 399
trial
serif::discretization::forms::FormDescriptor, 573
trial_order
serif::discretization::quadrature::Query, 821
trial_order_reduction
serif::discretization::forms::FormTraits< boundary_flux< TrialT, TestT > >, 575
serif::discretization::forms::FormTraits< divergence< TrialT, TestT > >, 577
serif::discretization::forms::FormTraits< force< TrialT, TestT > >, 578
serif::discretization::forms::FormTraits< FormT >, 574
serif::discretization::forms::FormTraits< mass< TrialT, TestT > >, 579
serif::discretization::forms::FormTraits< projection< TrialT, TestT > >, 581
serif::discretization::forms::FormTraits< source< TrialT, TestT > >, 582
serif::discretization::quadrature::Query, 821
trial_type
serif::discretization::forms::boundary_flux< TrialT, TestT >, 317
serif::discretization::forms::divergence< TrialT, TestT >, 430
serif::discretization::forms::force< TrialT, TestT >, 541
serif::discretization::forms::mass< TrialT, TestT >, 687
serif::discretization::forms::projection< TrialT, TestT >, 803
serif::discretization::forms::source< TrialT, TestT >, 916
try_evaluate
serif::eos::runtime::EOSView, 508
try_partial_derivative
serif::eos::runtime::EOSView, 509
try_prepare
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 371
try_prepare_nodal
serif::operators::backend::mfem, 188
try_prepare_point_border
serif::operators::backend::mfem, 188
try_prepare_validated
serif::operators::backend::mfem, 189
try_prepare_volume
serif::operators::backend::mfem, 189
tryRuntimePartialDerivative
serif::eos::runtime, 115
Type
serif::dimensions::QuantityAt< Index, QuantityList< Quantities... > >, 812
serif::dimensions::QuantityOf< QuantityValue< Quantity > >, 813
serif::eos::runtime::RuntimeRelationQuantities< relations::Relation< OutputT, InputTs... > >, 902
serif::geometry::deformation::DeformationWorkspaceType< Map, Linear >, 407
serif::geometry::deformation::DeformationWorkspaceType< Map, true >, 407
serif::geometry::surface::detail::SurfaceOperands< utils::dependencies::KeyList< Terms... > >, 947
serif::geometry::surface::detail::UniqueTerms< utils::types::TypeList< Terms... > >, 979
serif::utils::misc::concepts::Concat< types::TypeList< LhsTs... >, types::TypeList< RhsTs... >, Rest... >, 384
serif::utils::misc::concepts::Concat< types::TypeList< Ts... > >, 385
serif::utils::misc::concepts::Concat<>, 385
serif::utils::misc::concepts::EnumerateTypes< types::TypeList< TypeTs... >, Descriptor >, 502
serif::utils::misc::concepts::Filter< types::TypeList< Head, Tail... >, Predicate >, 536
serif::utils::misc::concepts::Filter< types::TypeList<>, Predicate >, 537
serif::utils::misc::concepts::Prepend< T, types::TypeList< Ts... > >, 784
serif::utils::types::DenseMatrixBase< T >, 408
serif::utils::types::DenseTensorBase< T, Rank >, 408
serif::utils::types::SparseMatrixBase< T >, 920
serif::utils::types::SparseTensorBase< T, Rank >, 920
serif::utils::types::VectorBase< T >, 1003
type
serif::discretization::forms::OperandResult< OperandT >, 735
serif::discretization::forms::OperationValue< 0, Operation::Gradient >, 741
serif::discretization::forms::OperationValue< 0, Operation::Value >, 741
serif::discretization::forms::OperationValue< 1, Operation::Curl >, 742
serif::discretization::forms::OperationValue< 1,

Operation::Divergence >, 742
serif::discretization::forms::OperationValue< 1, Operation::NormalTrace >, 743
serif::discretization::forms::OperationValue< 1, Operation::Value >, 744
serif::equations::Contributions< T, std::void_t< decltype(std::declval< const T & >().contributions()) > >, 394
serif::equations::Contributions< T, typename >, 393
serif::equations::ContributionSupport< T, std::void_t< typename T::Support > >, 395
serif::equations::ContributionSupport< T, typename >, 395
serif::equations::ContributionTuple< std::tuple< Ts... >, std::void_t< decltype(std::tuple_cat(std::declval< typename Contributions< std::remove_cvref_t< Ts > >::type >()...)) > >, 396
serif::equations::ContributionTuple< TupleT, typename >, 396
serif::equations::TestOperandTraits< T, std::void_t< typename T::TestOperand > >, 971
serif::equations::TestOperandTraits< T, typename >, 970
serif::operators::detail::AsKeyList< utils::types::TypeList< T... > >, 277
serif::operators::detail::Columns< utils::types::TypeList< O... > >, 355
serif::operators::detail::CouplingForm< Input, TestOperand >, 399
serif::operators::detail::Join< Lists >, 661
serif::utils::dependencies::detail::AppendBinding< Bindings< BindingsTs... >, BindingT >, 270
serif::utils::dependencies::detail::AppendKeys< ResultT, FirstT, Rest... >, 272
serif::utils::dependencies::detail::AppendKeys< ResultT, Keys >, 271
serif::utils::dependencies::detail::AppendUnique< KeyList< Keys... >, T >, 273
serif::utils::dependencies::detail::BoundType< Bindings< BindingTs... >, KeyT >::binding_tag< K, T >, 309
serif::utils::dependencies::detail::MergeKeyLists< ResultT, KeyList< Keys... >, Rest... >, 698
serif::utils::dependencies::detail::MergeKeyLists< ResultT, Lists >, 697
serif::utils::dependencies::detail::ResourceKeys< types::TypeList< Resources... > >, 878
serif::utils::dependencies::ResolveKeys< BindingsT, KeyList< KeyTs... > >, 874
type_index
serif::utils::dependencies::detail, 206
type_name
serif::meta, 154
unit_rotation
serif::equations::RigidRotation, 884
UNKNOWN_ERROR
serif::utils::errors, 209
unresolved_surface
serif::geometry::deformation, 122
Update
serif::utils::dependencies::IncrementalComputation< KeyT, ValueT, InputsT, FunctionT >, 630
used_fixed_order
serif::discretization::quadrature::Resolution, 870
UsedResources
serif::utils::dependencies::detail, 205
uses_tensor
serif::discretization::backend::mfem::BasisEvaluation, 298
serif::operators::backend::mfem::BasisEvaluation, 305
v_dot_n_scale
serif::mapping::FaceQuadratureContext, 532
vacuum
serif::geometry::deformation::DomainDeformation< SurfaceT, InteriorT, VacuumT >, 458
vacuum_attributes
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza 831
serif::geometry::deformation::backend::mfem::ReferenceDiscretiza 833
vacuum_element_attribute_ID
serif::utils::user_options::DomainMapperOptions, 474
vacuum_support
serif::geometry::deformation::RadialInterpolation, 829
VALID
serif::mapping, 150
valid
serif::discretization::domain::relation::validation::RelationValidati 850
serif::discretization::domain::schema::validation::RelationValidati 851
serif::discretization::domain::schema::validation::SchemaRelation 907
serif::discretization::domain::schema::validation::SchemaValidatio 909
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, true >, 375
serif::utils::dependencies::detail::ComputationCacheTraits< GraphT, ContextT, ErrorT, Valid >, 374

serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, Independent >, 378
serif::utils::dependencies::detail::ComputationGraphSchema< GraphT, ContextT, true >, 379
serif::utils::dependencies::detail::Invocation< ComputationT, ArgumentsT, typename >, 640
serif::utils::dependencies::detail::Invocation< ComputationT, types::TypeList< Arguments... >, std::void_t< decltype(std::declval< const ComputationT & >().evaluate(std::declval< const Arguments & >()...))> >, 642
serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, Available >, 870
serif::utils::dependencies::detail::ResolveComputation< ComputationT, BindingsT, true >, 871
serif::utils::dependencies::detail::ResolveGraph< GraphT, BindingsT, Position, true >, 873
serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, true >, 876
serif::utils::dependencies::detail::ResolveStep< GraphT, BindingsT, Position, ComputationT, Valid >, 875
serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >, 881
serif::utils::dependencies::detail::Result< T >, 880
validate
serif::discretization::backend::mfem::DofMap, 443
serif::discretization::backend::mfem::DofSelection, 448
serif::discretization::domain::schema::validation::RelationValidator< DomainBoundary< BoundaryT, DomainTs... > >, 853
serif::discretization::domain::schema::validation::RelationValidator< FullyConnected< DomainT > >, 853
serif::discretization::domain::schema::validation::RelationValidator< Inscribed< InnerT, OuterT > >, 854
serif::discretization::domain::schema::validation::SchemaValidator< SchemaT, relation::lists::RelationList< RelationTs... > >, 908
validate_companion_numbering
serif::geometry::deformation::backend::mfem, 132
validate_deformation_values
serif::geometry::deformation, 126
validate_disjoint_output
serif::geometry::deformation::detail, 136
validate_element_data
serif::mapping::DomainMapper, 473
validate_field_bindings
serif::mapping::GridFunctionMappingEvaluator, 618
validate_finite
serif::eos::models, 109
serif::utils::misc, 211
validate_greater_than
serif::eos::models, 109
serif::utils::misc, 211
validate_integral
serif::equations::AngularMomentum, 267
validate_less_than
serif::eos::models, 109
serif::utils::misc, 212
validate_nodal_discretization
serif::operators::backend::mfem, 189
validate_nodal_output
serif::operators::backend::mfem, 189
validate_nodal_selection
serif::operators::backend::mfem, 189
validate_nonnegativity
serif::eos::models, 109
serif::utils::misc, 212
validate_point
serif::mapping::VolumeMappingCache, 1016
validate_point_border_outputs
serif::operators::backend::mfem, 190
validate_relation_request
serif::eos::runtime::EOSView, 509
validate_scalar_reference_space
serif::geometry::deformation::backend::mfem, 132
validate_schema
serif::discretization::domain::schema::validation, 91
validate_validator
serif::discretization::backend::mfem::DofMap, 443
validate_validator_data
serif::operators::backend::mfem, 190
validate_validator_discretization
serif::operators::backend::mfem, 190
validationValidator< serif::discretization::quadrature::QuadratureOptions, 808
validation_integral
serif::operators::backend::mfem::ElementData< OperandCount >, 487
validation_integrand
serif::equations::AngularMomentum, 268
Value
serif::equations::gravity_contributions, 118
serif::operators::backend::mfem::NodalDiscretization< EquationT, SchemaT >, 715
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 730

serif::utils::dependencies::detail::Result< std::expected< ValueT, ErrorT > >, 881
serif::utils::dependencies::detail::Result< T >, 879
value
serif::dimensions::PartialDerivative< OutputQuantity, InputQuantity >, 751
serif::dimensions::QuantityValue< Quantity >, 815
serif::dimensions::runtime::RuntimeQuantityValue, 900
serif::discretization::backend::mfem, 70
serif::discretization::functional_space, 99
serif::discretization::functional_space::Rank< R >, 830
serif::discretization::quadrature::backend::mfem::MFEMOperandEvaluator< utils::types::TypeList< OperandTs... > >, 704
serif::geometry::surface::SurfaceEOSResponse< Formulation >, 945
serif::mapping::DMReferenceDisplacementSample, 433
serif::utils::CompensatedSum, 357
serif::utils::dependencies::detail::BorrowedValue< T >, 315
serif::utils::dependencies::detail::CachedResult< ValueT, InputCount >, 333
serif::utils::dependencies::detail::HasBinding< Bindings< BindingTs... >, KeyT >, 621
serif::utils::dependencies::detail::OwnedValue< T >, 748
value_and_gradient
serif::discretization::backend::mfem, 70
values
serif::equations::OperandValues< utils::types::TypeList< Operands... >, OutputT >, 738
serif::geometry::surface::BoundaryValues< Term >, 328
serif::operators::backend::mfem, 191
serif::operators::backend::mfem::CoefficientView< BindingT >, 354
serif::operators::backend::mfem::MappingCoefficients, 676
serif::operators::backend::mfem::SelectedNodalValues, 911
serif::operators::detail::ContributionChildren< EquationT >, 390
Vector
serif::equations::RigidRotation, 883
serif::operators::backend::mfem, 171
vector_copies
serif::operators::backend::mfem::OperandEvaluator< OperandT >, 733
vector_dimension
serif::geometry::deformation::backend::mfem::ReferenceSpaceSnap 838
vector_term
serif::discretization::blocks::displacement, 425
VectorValue
serif::discretization::forms, 94
VelocityValue
serif::dimensions, 67
version_index
serif::utils::dependencies::ComputationCache< GraphT, ContextT, ErrorT >, 371
versions
serif::utils::dependencies::detail::ComputationCacheStorage< GraphT, BindingsT, types::TypeList< Resources... >, std::index_sequence< Indices... > >, 374
vertices
serif::geometry::deformation::backend::mfem::ReferenceMeshSnap 836
view
serif::utils::types::ct_error_string< N >, 400
serif::utils::types::ctstring< N >, 401
views.cpp
SERiFErrorCode, 1038
virial
serif::discretization::quadrature::QuadratureOptions, 809
visit_contributions
serif::operators::detail, 193
visitedElementCount
serif::discretization::domain::relation::validation::RelationValidati 386
serif::discretization::domain::schema::validation::RelationValidati 387
volume_height
serif::operators::backend::mfem, 190
volume_residual
serif::operators::backend::mfem, 190
volume_width
serif::operators::backend::mfem, 191
volume_work_report
serif::operators::backend::mfem, 191
VolumeCache
serif::operators::backend::mfem, 171
VolumeCoefficients
serif::operators::backend::mfem, 171
VolumeDiscretization
serif::operators::backend::mfem, 191
VolumeEquationOf
serif::operators::backend::mfem, 171
VolumeInputs
serif::operators::backend::mfem, 172
VolumeMesh
serif::operators::backend::mfem, 172

VolumeOperandResult
serif::operators::backend::mfem, 172
VolumeValue
serif::dimensions, 67

warning.hpp
DISABLE_WARNING, 1369
weight
serif::mapping::DMFixedMeshGeometry, 431
weight_variation
serif::mapping::VolumeMappingVariation, 1019
weighted_field_derivatives
serif::operators::backend::mfem::ElementData<
OperandCount >, 488
weighted_position_derivative
serif::operators::backend::mfem::ElementData<
OperandCount >, 488
weighted_residual
serif::operators::backend::mfem::ElementData<
OperandCount >, 488
WHITE
serif::utils::misc::terminal, 217
width
serif::discretization::backend::mfem::BasisEvaluation,
298
serif::operators::backend::mfem::BasisEvaluation,
305
workspace
serif::operators::backend::mfem::VolumeScratch,
1023

YELLOW
serif::utils::misc::terminal, 217