

Interface Reconstruction Library (IRL)

Generated by Doxygen 1.8.15

Contents

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	5
2.1 Class Hierarchy	5
3 Class Index	29
3.1 Class List	29
4 File Index	49
4.1 File List	49
5 Namespace Documentation	55
5.1 f_bytebuffer_class Module Reference	55
5.1.1 Detailed Description	55
5.2 f_cappeddodecahedron_class Module Reference	56
5.2.1 Detailed Description	56
5.3 f_cappeddodecahedron_doubles3_class Module Reference	56
5.3.1 Detailed Description	57
5.4 f_constants Module Reference	57
5.4.1 Detailed Description	58
5.5 f_cutpolygon Module Reference	58
5.5.1 Detailed Description	58
5.6 f_definedtypes Module Reference	59
5.6.1 Detailed Description	59

5.7 f_dividedpolygon_class Module Reference	59
5.7.1 Detailed Description	60
5.8 f_dodecahedron_class Module Reference	61
5.8.1 Detailed Description	61
5.9 f_elviraneighborhood_class Module Reference	61
5.9.1 Detailed Description	62
5.10 f_geometriccuttinghelpers Module Reference	62
5.10.1 Detailed Description	62
5.11 f_getvolumemoments Module Reference	62
5.11.1 Detailed Description	64
5.12 f_listedvm_vman_class Module Reference	64
5.12.1 Detailed Description	64
5.13 f_localizedseparatorlink_class Module Reference	65
5.13.1 Detailed Description	65
5.14 f_localizerlink_class Module Reference	65
5.14.1 Detailed Description	66
5.15 f_lviraneighborhood_rectangularcuboid_class Module Reference	66
5.15.1 Detailed Description	67
5.16 f_objectallocationserver_localizedseparatorlink_class Module Reference	67
5.16.1 Detailed Description	67
5.17 f_objectallocationserver_localizerlink_class Module Reference	68
5.17.1 Detailed Description	68
5.18 f_objectallocationserver_planarlocalizer_class Module Reference	68
5.18.1 Detailed Description	69
5.19 f_objectallocationserver_planarseparator_class Module Reference	69
5.19.1 Detailed Description	69
5.20 f_planarlocalizer_class Module Reference	69
5.20.1 Detailed Description	70
5.21 f_planarseparator_class Module Reference	70
5.21.1 Detailed Description	71

5.22 f_polygon_class Module Reference	71
5.22.1 Detailed Description	72
5.23 f_polyhedron24_class Module Reference	72
5.23.1 Detailed Description	73
5.24 f_polyhedron24_doubles3_class Module Reference	73
5.24.1 Detailed Description	74
5.25 f_r2pneighborhood_rectangularcuboid_class Module Reference	74
5.25.1 Detailed Description	75
5.26 f_reconstructioninterface Module Reference	75
5.26.1 Detailed Description	77
5.27 f_rectangularcuboid_class Module Reference	77
5.27.1 Detailed Description	77
5.28 f_sepvm_class Module Reference	77
5.28.1 Detailed Description	78
5.29 f_sepvm_doubles3_class Module Reference	78
5.29.1 Detailed Description	79
5.30 f_serializer Module Reference	79
5.30.1 Detailed Description	80
5.31 f_tagged_accumlistedvm_vman_class Module Reference	80
5.31.1 Detailed Description	81
5.32 f_tagged_accumvm_sepvm_class Module Reference	81
5.32.1 Detailed Description	82
5.33 f_tagged_accumvm_vm_class Module Reference	82
5.33.1 Detailed Description	83
5.34 f_tet_class Module Reference	83
5.34.1 Detailed Description	83
5.35 f_tri_class Module Reference	84
5.35.1 Detailed Description	85
5.36 f_vman_class Module Reference	85
5.36.1 Detailed Description	85

5.37 f_volumefractionmatching Module Reference	85
5.37.1 Detailed Description	86
5.38 IRL Namespace Reference	86
5.38.1 Detailed Description	105
5.38.2 Function Documentation	106
5.38.2.1 __attribute__()	106
5.38.2.2 checkPureObject()	106
5.38.2.3 crossProduct()	107
5.38.2.4 crossProductNormalized()	107
5.38.2.5 cutPlaneByHexahedron()	107
5.38.2.6 cutPolygonByPlane()	108
5.38.2.7 dotProduct()	108
5.38.2.8 findDistanceOnePlane()	109
5.38.2.9 getGeometricCaseId()	109
5.38.2.10 getPlanePolygonFromReconstruction()	109
5.38.2.11 getReconstructionSurfaceArea()	110
5.38.2.12 isPlaneIntersectingCell()	111
5.38.2.13 isPtInternal()	111
5.38.2.14 magnitude()	112
5.38.2.15 rotateNormalOntoNormal()	112
5.38.2.16 scalarTripleProduct()	112
5.38.2.17 setDistanceToMatchVolumeFractionPartialFill()	113
5.38.2.18 setVolumeFractionBounds()	113
5.38.2.19 setVolumeFractionTolerance()	114
5.38.2.20 signedDistanceToPolygonVertices()	114
5.38.2.21 signedDistanceToVertices()	114
5.38.2.22 sortAscendingBasedOnOtherArray()	115
5.38.2.23 sortDescendingBasedOnOtherArray()	116
5.38.2.24 squaredMagnitude()	116
5.38.3 Variable Documentation	117
5.38.3.1 C_CUTTING_METHOD	117
5.38.3.2 unit_cell	117
5.39 irl_fortran_interface Module Reference	117
5.39.1 Detailed Description	117

6 Class Documentation	119
6.1 IRL::AccumulatedListedVolumeMoments< VolumeMomentsType > Class Template Reference	119
6.2 IRL::AccumulatedVolumeMoments< VolumeMomentsType > Class Template Reference	121
6.2.1 Detailed Description	122
6.3 IRL::AccumulateIntoCollection Struct Reference	122
6.4 IRL::AccumulateIntoScalar Struct Reference	123
6.5 IRL::AccumulateWrapper< WrappedType > Class Template Reference	123
6.5.1 Detailed Description	124
6.6 IRL::Accumulator< ObjectType > Class Template Reference	124
6.6.1 Detailed Description	125
6.7 f_lviraneighborhood_rectangularcuboid_class::addmember Interface Reference	125
6.8 f_r2pneighborhood_rectangularcuboid_class::addmember Interface Reference	126
6.9 f_planarseparator_class::addplane Interface Reference	126
6.10 f_planarlocalizer_class::addplane Interface Reference	126
6.11 f_cappeddodecahedron_doubles3_class::adjustcaptomatchvolume Interface Reference	126
6.12 f_cappeddodecahedron_class::adjustcaptomatchvolume Interface Reference	127
6.13 f_polyhedron24_class::adjustcaptomatchvolume Interface Reference	127
6.14 f_polyhedron24_doubles3_class::adjustcaptomatchvolume Interface Reference	127
6.15 IRL::AdvectionPlaneReconstruction Class Reference	127
6.16 IRL::AdvectionPlaneReconstructionDebug Class Reference	128
6.17 f_listedvm_vman_class::append Interface Reference	129
6.18 f_tagged_accumlistedvm_vman_class::append Interface Reference	129
6.19 short_alloc::arena< N, alignment > Class Template Reference	129
6.20 IRL::BFGS< OptimizingClass, kParameters > Class Template Reference	130
6.20.1 Detailed Description	131
6.21 IRL::Bisection< OptimizingClass > Class Template Reference	131
6.21.1 Detailed Description	132
6.21.2 Member Function Documentation	132
6.21.2.1 solve() [1/2]	132
6.21.2.2 solve() [2/2]	133

6.22 IRL::BlockObjectAllocation< ObjectType > Class Template Reference	133
6.23 IRL::BrentsMethod< OptimizingClass > Class Template Reference	134
6.23.1 Detailed Description	134
6.23.2 Member Function Documentation	135
6.23.2.1 solve() [1/2]	135
6.23.2.2 solve() [2/2]	135
6.24 IRL::ByteBuffer Class Reference	136
6.25 f_bytebuffer_class::bytebuffer_type Type Reference	137
6.26 f_bytebuffer_class::c_bytebuffer Type Reference	137
6.27 c_ByteBuffer Struct Reference	138
6.28 c_CappedDodecahedron Struct Reference	138
6.29 f_cappeddodecahedron_class::c_cappeddodecahedron Type Reference	139
6.30 c_CappedDodecahedron_doubles3 Struct Reference	139
6.31 f_cappeddodecahedron_doubles3_class::c_cappeddodecahedron_doubles3 Type Reference	139
6.32 c_DividedPolygon Struct Reference	140
6.33 f_dividedpolygon_class::c_dividedpolygon Type Reference	140
6.34 f_dodecahedron_class::c_dodecahedron Type Reference	140
6.35 c_Dodecahedron Struct Reference	140
6.36 c_ELVIRANeighborhood Struct Reference	141
6.37 f_elviraneighborhood_class::c_elviraneighborhood Type Reference	142
6.38 c_ListedVM_VMAN Struct Reference	142
6.39 f_listedvm_vman_class::c_listedvm_vman Type Reference	143
6.40 c_LocalizedSeparatorLink Struct Reference	143
6.41 f_localizedseparatorlink_class::c_localizedseparatorlink Type Reference	143
6.42 c_LocalizerLink Struct Reference	144
6.43 f_localizerlink_class::c_localizerlink Type Reference	144
6.44 c_LVIRANeighborhood_RectangularCuboid Struct Reference	145
6.45 f_lviraneighborhood_rectangularcuboid_class::c_lviraneighborhood_rectangularcuboid Type Reference	145
6.46 c_ObjectAllocationServer_LocalizedSeparatorLink Struct Reference	146

6.47 f_objectallocationserver_localizedseparatorlink_class::c_objectallocationserver_localizedseparatorlink Type Reference	146
6.48 f_objectallocationserver_localizerlink_class::c_objectallocationserver_localizerlink Type Reference	146
6.49 c_ObjectAllocationServer_LocalizerLink Struct Reference	147
6.50 f_objectallocationserver_planarlocalizer_class::c_objectallocationserver_planarlocalizer Type Reference	147
6.51 c_ObjectAllocationServer_PlanarLocalizer Struct Reference	148
6.52 c_ObjectAllocationServer_PlanarSeparator Struct Reference	149
6.53 f_objectallocationserver_planarseparator_class::c_objectallocationserver_planarseparator Type Reference	150
6.54 f_planarlocalizer_class::c_planarlocalizer Type Reference	150
6.55 c_PlanarLocalizer Struct Reference	150
6.56 f_planarseparator_class::c_planarseparator Type Reference	151
6.57 c_PlanarSeparator Struct Reference	151
6.58 f_polygon_class::c_polygon Type Reference	152
6.59 c_Polygon Struct Reference	152
6.60 f_polyhedron24_class::c_polyhedron24 Type Reference	152
6.61 c_Polyhedron24 Struct Reference	153
6.62 f_polyhedron24_doubles3_class::c_polyhedron24_doubles3 Type Reference	153
6.63 c_Polyhedron24_doubles3 Struct Reference	153
6.64 c_R2PNeighborhood_RectangularCuboid Struct Reference	154
6.65 f_r2pneighborhood_rectangularcuboid_class::c_r2pneighborhood_rectangularcuboid Type Reference	154
6.66 c_RectangularCuboid Struct Reference	155
6.67 f_rectangularcuboid_class::c_rectangularcuboid Type Reference	155
6.68 c_SepVM Struct Reference	155
6.69 f_sepvm_class::c_sepvm Type Reference	156
6.70 c_SepVM_doubles3 Struct Reference	156
6.71 f_sepvm_doubles3_class::c_sepvm_doubles3 Type Reference	157
6.72 c_Tagged_AccumListedVM_VMAN Struct Reference	157
6.73 f_tagged_accumlistedvm_vman_class::c_tagged_accumlistedvm_vman Type Reference	157
6.74 f_tagged_accumvm_sepvm_class::c_tagged_accumvm_sepvm Type Reference	158
6.75 c_Tagged_AccumVM_SepVM Struct Reference	158

6.76 f_tagged_accumvm_vm_class::c_tagged_accumvm_vm Type Reference	158
6.77 c_Tagged_AccumVM_VM Struct Reference	159
6.78 f_tet_class::c_tet Type Reference	159
6.79 c_Tet Struct Reference	159
6.80 f_tri_class::c_tri Type Reference	160
6.81 c_Tri Struct Reference	160
6.82 f_vman_class::c_vman Type Reference	160
6.83 c_VMAN Struct Reference	161
6.84 f_tri_class::calculateandsetplaneofexistence Interface Reference	161
6.85 f_polygon_class::calculateandsetplaneofexistence Interface Reference	162
6.86 f_dividedpolygon_class::calculateandsetplaneofexistence Interface Reference	162
6.87 f_tri_class::calculatecentroid Interface Reference	162
6.88 f_polygon_class::calculatecentroid Interface Reference	162
6.89 f_polygon_class::calculatenearestptonsurface Interface Reference	163
6.90 f_tri_class::calculatenormal Interface Reference	163
6.91 f_polygon_class::calculatenormal Interface Reference	163
6.92 f_dividedpolygon_class::calculatenormal Interface Reference	163
6.93 f_tri_class::calculatesign Interface Reference	164
6.94 f_polygon_class::calculatesign Interface Reference	164
6.95 f_dividedpolygon_class::calculatesign Interface Reference	164
6.96 f_dividedpolygon_class::calculatesurfacearea Interface Reference	164
6.97 f_tri_class::calculatevolume Interface Reference	165
6.98 f_polygon_class::calculatevolume Interface Reference	165
6.99 f_rectangularcuboid_class::calculatevolume Interface Reference	165
6.100 f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_type Type Reference	165
6.101 f_cappeddodecahedron_class::cappeddodecahedron_type Type Reference	166
6.102 IRL::CappedDodecahedronSpecialization< Derived, VertexType > Class Template Reference	167
6.103 IRL::CellCollection< CellGroupType > Class Template Reference	168
6.103.1 Detailed Description	169
6.104 IRL::CellGroupedMoments< CellType, ContainedMomentsType > Class Template Reference	169

6.104.1 Detailed Description	170
6.105 IRL::Centroid2D_Functor Class Reference	171
6.106 IRL::Centroid3D_Functor Class Reference	172
6.107 IRL::ChainedBlockStorage< ContainedType, kBlockSize > Class Template Reference	173
6.108 f_listedvmm_vman_class::clear Interface Reference	174
6.109 f_tagged_accumlistedvmm_vman_class::clear Interface Reference	174
6.110 IRL::Collection< ObjectType > Class Template Reference	174
6.110.1 Detailed Description	175
6.111 IRL::ConcaveBoxSpecialization< Derived, VertexType > Class Template Reference	176
6.112 IRL::ConstIteratorThroughBracketOperator< ContainerType > Class Template Reference	177
6.113 f_tet_class::construct Interface Reference	178
6.114 f_tri_class::construct Interface Reference	178
6.115 f_cappeddodecahedron_class::construct Interface Reference	178
6.116 f_dodecahedron_class::construct Interface Reference	179
6.117 f_cappeddodecahedron_doubles3_class::construct Interface Reference	179
6.118 f_polygon_class::construct Interface Reference	179
6.119 f_polyhedron24_class::construct Interface Reference	179
6.120 f_polyhedron24_doubles3_class::construct Interface Reference	180
6.121 f_dividedpolygon_class::construct Interface Reference	180
6.122 f_rectangularcuboid_class::construct Interface Reference	180
6.123 f_sepvm_class::construct Interface Reference	180
6.124 f_rectangularcuboid_class::construct_2pt Interface Reference	181
6.125 f_dividedpolygon_class::constructfrompolygon Interface Reference	181
6.126 IRL::ContinueDividingVolumeByPlanarReconstruction< HowToCutSimplexByNextPlane, HowToBranchEarlyIfNoIntersection, HowToGenerateAndHandleSimplicesFromVolumeUnderPlane, HowToGenerateAndHandleSimplicesFromVolumeAbovePlane > Class Template Reference	181
6.127 IRL::ContinueReducingVolumeToBeInternalToReconstruction Struct Reference	182
6.128 f_planarseparator_class::copy Interface Reference	182
6.129 IRL::CutSimplexByNextPlaneAccessedByIndex Struct Reference	182
6.130 IRL::CutTetVertices< VertexType > Class Template Reference	182
6.131 IRL::CutTriangleVerticesAndPlane< VertexType > Class Template Reference	183

6.132 f_bytebuffer_class::dataptr Interface Reference	184
6.133 IRL::DecomposedPolygonVertexStorage< VertexType > Class Template Reference	184
6.134 IRL::DecomposedPolyhedronVertexStorage< VertexType > Class Template Reference	186
6.135 IRL::DecomposedPolytopeVertexStorage< VertexType > Class Template Reference	187
6.136 IRL::DelayedExpandableVertexAccess< Derived, VertexType, kMaskedVertices > Class Template Reference	188
6.137 f_dividedpolygon_class::dividedpolygon_type Type Reference	189
6.138 IRL::DividedPolygonSpecialization< Derived, VertexType > Class Template Reference	189
6.138.1 Detailed Description	190
6.139 IRL::DivideSimplexByPlanarReconstruction< HowToHandleVolumeCompletelyInternalToPlanar← Reconstruction, HowToContinueDividingVolumeByPlanarReconstruction > Class Template Reference	191
6.140 f_dodecahedron_class::dodecahedron_type Type Reference	192
6.141 IRL::DodecahedronSpecialization< Derived, VertexType > Class Template Reference	192
6.142 IRL::DoesNotHaveACollection< ReturnType > Struct Template Reference	193
6.143 IRL::DoesNotHaveALocalizer< ReconstructionType > Struct Template Reference	193
6.144 IRL::DoesNotHaveANestedType< ReturnType > Struct Template Reference	194
6.145 IRL::DoesNotHaveAReconstructionLink< ReconstructionType > Struct Template Reference	194
6.145.1 Member Data Documentation	194
6.145.1.1 value	194
6.146 IRL::DoesNotHaveASeparator< ReconstructionType > Struct Template Reference	194
6.147 IRL::doNoEarlyBranching Struct Reference	195
6.148 IRL::earlyBranchBothFullyBelowAndFullyAboveSimplicesLink Struct Reference	195
6.149 IRL::earlyBranchBothFullyBelowAndFullyAboveSimplicesNoLink Struct Reference	195
6.150 IRL::ELVIRA_2D Class Reference	196
6.151 IRL::ELVIRA_3D Class Reference	197
6.152 IRL::ELVIRADebug< BaseELVIRA > Class Template Reference	199
6.153 IRL::ELVIRADebug< ELVIRA_2D > Class Template Reference	199
6.154 IRL::ELVIRADebug< ELVIRA_3D > Class Template Reference	200
6.155 IRL::ELVIRANeighborhood Class Reference	201
6.155.1 Detailed Description	202
6.156 f_elviraneighborhood_class::elviraneighborhood_type Type Reference	203

6.157 f_r2pneighborhood_rectangularcuboid_class::emptyneighborhood Interface Reference	203
6.158 f_lviraneighborhood_rectangularcuboid_class::emptyneighborhood Interface Reference	204
6.159 f_listedvmm_vman_class::erase Interface Reference	204
6.160 IRL::ExpandableDividedPolygon< VertexType > Class Template Reference	204
6.161 IRL::ExpandablePolygon< VertexType > Class Template Reference	205
6.162 IRL::ExpandablePtList< VertexType > Class Template Reference	206
6.163 IRL::Expr< E > Struct Template Reference	208
6.164 f_bytebuffer_class::F_ByteBuffer_dataPtr Interface Reference	208
6.165 f_bytebuffer_class::F_ByteBuffer_delete Interface Reference	208
6.166 f_bytebuffer_class::F_ByteBuffer_getSize Interface Reference	208
6.167 f_bytebuffer_class::F_ByteBuffer_new Interface Reference	209
6.168 f_bytebuffer_class::F_ByteBuffer_resetBufferPointer Interface Reference	209
6.169 f_bytebuffer_class::F_ByteBuffer_setSize Interface Reference	209
6.170 f_cappeddodecahedron_class::F_CappedDodecahedron_adjustCapToMatchVolume Interface Reference	209
6.171 f_cappeddodecahedron_class::F_CappedDodecahedron_construct Interface Reference	210
6.172 f_cappeddodecahedron_class::F_CappedDodecahedron_delete Interface Reference	210
6.173 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_adjustCapToMatchVolume Interface Reference	210
6.174 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_construct Interface Reference	210
6.175 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_delete Interface Reference	211
6.176 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getBoundingPts Interface Reference	211
6.177 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getData Interface Reference	211
6.178 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getPt Interface Reference	211
6.179 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_new Interface Reference	212
6.180 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_setData Interface Reference	212
6.181 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_setPt Interface Reference	212

6.182 f_cappeddodecahedron_class::F_CappedDodecahedron_getBoundingPts Interface Reference	212
6.183 f_cappeddodecahedron_class::F_CappedDodecahedron_getPt Interface Reference	213
6.184 f_cappeddodecahedron_class::F_CappedDodecahedron_new Interface Reference	213
6.185 f_constants::F_Constants_setMinimumSurfaceAreaToTrack Interface Reference	213
6.186 f_constants::F_Constants_setMinimumVolumeToTrack Interface Reference	213
6.187 f_constants::F_Constants_setVolumeFractionBounds Interface Reference	214
6.188 f_constants::F_Constants_setVolumeFractionToleranceForDistanceFinding Interface Reference	214
6.189 f_dividedpolygon_class::F_DividedPolygon_calculateAndSetPlaneOfExistence Interface Reference	214
6.190 f_dividedpolygon_class::F_DividedPolygon_calculateNormal Interface Reference	214
6.191 f_dividedpolygon_class::F_DividedPolygon_calculateSign Interface Reference	215
6.192 f_dividedpolygon_class::F_DividedPolygon_calculateSurfaceArea Interface Reference	215
6.193 f_dividedpolygon_class::F_DividedPolygon_construct Interface Reference	215
6.194 f_dividedpolygon_class::F_DividedPolygon_constructFromPolygon Interface Reference	215
6.195 f_dividedpolygon_class::F_DividedPolygon_delete Interface Reference	216
6.196 f_dividedpolygon_class::F_DividedPolygon_getBoundingPts Interface Reference	216
6.197 f_dividedpolygon_class::F_DividedPolygon_getLocalizer Interface Reference	216
6.198 f_dividedpolygon_class::F_DividedPolygon_getNumberOfPts Interface Reference	216
6.199 f_dividedpolygon_class::F_DividedPolygon_getNumberOfSimplicesInDecomposition Interface Reference	217
6.200 f_dividedpolygon_class::F_DividedPolygon_getPlaneOfExistence Interface Reference	217
6.201 f_dividedpolygon_class::F_DividedPolygon_getPt Interface Reference	217
6.202 f_dividedpolygon_class::F_DividedPolygon_getSimplexFromDecomposition Interface Reference	217
6.203 f_dividedpolygon_class::F_DividedPolygon_new Interface Reference	218
6.204 f_dividedpolygon_class::F_DividedPolygon_printToScreen Interface Reference	218
6.205 f_dividedpolygon_class::F_DividedPolygon_resetCentroid Interface Reference	218
6.206 f_dividedpolygon_class::F_DividedPolygon_reversePtOrdering Interface Reference	218
6.207 f_dividedpolygon_class::F_DividedPolygon_setPlaneOfExistence Interface Reference	219
6.208 f_dividedpolygon_class::F_DividedPolygon_zeroPolygon Interface Reference	219
6.209 f_dodecahedron_class::F_Dodecahedron_construct Interface Reference	219
6.210 f_dodecahedron_class::F_Dodecahedron_delete Interface Reference	219
6.211 f_dodecahedron_class::F_Dodecahedron_getBoundingPts Interface Reference	220

6.212 f_dodecahedron_class::F_Dodecahedron_new Interface Reference	220
6.213 f_elviraneighborhood_class::F_ELVIRANeighborhood_delete Interface Reference	220
6.214 f_elviraneighborhood_class::F_ELVIRANeighborhood_new Interface Reference	220
6.215 f_elviraneighborhood_class::F_ELVIRANeighborhood_setMember Interface Reference	221
6.216 f_elviraneighborhood_class::F_ELVIRANeighborhood_setSize Interface Reference	221
6.217 f_cutpolygon::F_getPlanePolygonFromReconstruction_RC_DivPoly Interface Reference	221
6.218 f_cutpolygon::F_getPlanePolygonFromReconstruction_RC_Poly Interface Reference	221
6.219 f_cutpolygon::F_getReconstructionSurfaceArea_RC Interface Reference	222
6.220 f_getvolumemoments::F_GNVM_CD_By_LSL_For_SVM Interface Reference	222
6.221 f_getvolumemoments::F_GNVM_CD_By_LSL_For_TagAccumVM_SVM Interface Reference	222
6.222 f_getvolumemoments::F_GNVM_CDWD3_By_LSL_For_SVMAD3 Interface Reference	222
6.223 f_getvolumemoments::F_GNVM_D_By_LSL_For_SVM Interface Reference	223
6.224 f_getvolumemoments::F_GNVM_D_By_LSL_For_TagAccumVM_SVM Interface Reference	223
6.225 f_getvolumemoments::F_GNVM_D_By_PS_For_SVM Interface Reference	223
6.226 f_getvolumemoments::F_GNVM_P24_By_LSL_For_SVM Interface Reference	223
6.227 f_getvolumemoments::F_GNVM_P24WD3_By_LSL_For_SVMAD3 Interface Reference	224
6.228 f_getvolumemoments::F_GNVM_Poly_By_PL_For_V Interface Reference	224
6.229 f_getvolumemoments::F_GNVM_RC_By_PS_For_SVM Interface Reference	224
6.230 f_getvolumemoments::F_GNVM_RC_By_PS_For_V Interface Reference	224
6.231 f_getvolumemoments::F_GNVM_Tet_By_LSL_For_SVM Interface Reference	225
6.232 f_getvolumemoments::F_GNVM_Tri_By_LL_For_TagAVM_VM Interface Reference	225
6.233 f_getvolumemoments::F_GNVM_Tri_By_PL_For_V Interface Reference	225
6.234 f_getvolumemoments::F_GVM_CD_By_LSL_For_SVM Interface Reference	225
6.235 f_getvolumemoments::F_GVM_D_By_LSL_For_SVM Interface Reference	226
6.236 f_getvolumemoments::F_GVM_P24_By_LSL_For_SVM Interface Reference	226
6.237 f_getvolumemoments::F_GVM_setMethod Interface Reference	226
6.238 f_getvolumemoments::F_GVM_Tri_By_LL_For_TagALVM_VMAN Interface Reference	226
6.239 f_geometriccuttinghelpers::F_isPtInternal_PL Interface Reference	227
6.240 f_geometriccuttinghelpers::F_isPtInternal_PS Interface Reference	227
6.241 f_listedvm_vman_class::F_ListedVM_VMAN_append Interface Reference	227

6.242 f_listedvm_vman_class::F_ListedVM_VMAN_clear Interface Reference	227
6.243 f_listedvm_vman_class::F_ListedVM_VMAN_delete Interface Reference	228
6.244 f_listedvm_vman_class::F_ListedVM_VMAN_erase Interface Reference	228
6.245 f_listedvm_vman_class::F_ListedVM_VMAN_getMoments Interface Reference	228
6.246 f_listedvm_vman_class::F_ListedVM_VMAN_getSize Interface Reference	228
6.247 f_listedvm_vman_class::F_ListedVM_VMAN_new Interface Reference	229
6.248 f_listedvm_vman_class::F_ListedVM_VMAN_zeroNormalComponent Interface Reference	229
6.249 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_delete Interface Reference	229
6.250 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_getId Interface Reference	229
6.251 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_new Interface Reference	230
6.252 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_newFromObjectAllocationServer Interface Reference	230
6.253 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivity Interface Reference	230
6.254 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivityNull Interface Reference	230
6.255 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setId Interface Reference	231
6.256 f_localizerlink_class::F_LocalizerLink_delete Interface Reference	231
6.257 f_localizerlink_class::F_LocalizerLink_getId Interface Reference	231
6.258 f_localizerlink_class::F_LocalizerLink_new Interface Reference	231
6.259 f_localizerlink_class::F_LocalizerLink_newFromObjectAllocationServer Interface Reference	232
6.260 f_localizerlink_class::F_LocalizerLink_setEdgeConnectivity Interface Reference	232
6.261 f_localizerlink_class::F_LocalizerLink_setEdgeConnectivityNull Interface Reference	232
6.262 f_localizerlink_class::F_LocalizerLink_setId Interface Reference	232
6.263 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_add← Member Interface Reference	233
6.264 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_delete Interface Reference	233
6.265 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_← emptyNeighborhood Interface Reference	233
6.266 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_new Interface Reference	233
6.267 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_set← CenterOfStencil Interface Reference	234

6.268 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_set← Member Interface Reference	234
6.269 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_set← Size Interface Reference	234
6.270 f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_Localized← SeparatorLink_delete Interface Reference	234
6.271 f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_Localized← SeparatorLink_new Interface Reference	235
6.272 f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_delete Inter- face Reference	235
6.273 f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_new Inter- face Reference	235
6.274 f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_delete Interface Reference	235
6.275 f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_new In- terface Reference	236
6.276 f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_PlanarSeparator_← delete Interface Reference	236
6.277 f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_PlanarSeparator_new Interface Reference	236
6.278 f_planarlocalizer_class::F_PlanarLocalizer_addPlane Interface Reference	236
6.279 f_planarlocalizer_class::F_PlanarLocalizer_delete Interface Reference	237
6.280 f_planarlocalizer_class::F_PlanarLocalizer_new Interface Reference	237
6.281 f_planarlocalizer_class::F_PlanarLocalizer_newFromObjectAllocationServer Interface Reference . .	237
6.282 f_planarlocalizer_class::F_PlanarLocalizer_printToScreen Interface Reference	237
6.283 f_planarlocalizer_class::F_PlanarLocalizer_setFromRectangularCuboid Interface Reference . . .	238
6.284 f_planarlocalizer_class::F_PlanarLocalizer_setNumberOfPlanes Interface Reference	238
6.285 f_planarlocalizer_class::F_PlanarLocalizer_setPlane Interface Reference	238
6.286 f_planarseparator_class::F_PlanarSeparator_addPlane Interface Reference	238
6.287 f_planarseparator_class::F_PlanarSeparator_copy Interface Reference	239
6.288 f_planarseparator_class::F_PlanarSeparator_delete Interface Reference	239
6.289 f_planarseparator_class::F_PlanarSeparator_getNumberOfPlanes Interface Reference	239
6.290 f_planarseparator_class::F_PlanarSeparator_getPlane Interface Reference	239
6.291 f_planarseparator_class::F_PlanarSeparator_isFlipped Interface Reference	240
6.292 f_planarseparator_class::F_PlanarSeparator_new Interface Reference	240

6.293 f_planarseparator_class::F_PlanarSeparator_newFromObjectAllocationServer Interface Reference	240
6.294 f_planarseparator_class::F_PlanarSeparator_printToScreen Interface Reference	240
6.295 f_planarseparator_class::F_PlanarSeparator_setNumberOfPlanes Interface Reference	241
6.296 f_planarseparator_class::F_PlanarSeparator_setPlane Interface Reference	241
6.297 f_polygon_class::F_Polygon_calculateAndSetPlaneOfExistence Interface Reference	241
6.298 f_polygon_class::F_Polygon_calculateCentroid Interface Reference	241
6.299 f_polygon_class::F_Polygon_calculateNearestPtOnSurface Interface Reference	242
6.300 f_polygon_class::F_Polygon_calculateNormal Interface Reference	242
6.301 f_polygon_class::F_Polygon_calculateSign Interface Reference	242
6.302 f_polygon_class::F_Polygon_calculateVolume Interface Reference	242
6.303 f_polygon_class::F_Polygon_construct Interface Reference	243
6.304 f_polygon_class::F_Polygon_delete Interface Reference	243
6.305 f_polygon_class::F_Polygon_getBoundingPts Interface Reference	243
6.306 f_polygon_class::F_Polygon_getLocalizer Interface Reference	243
6.307 f_polygon_class::F_Polygon_getNumberOfPts Interface Reference	244
6.308 f_polygon_class::F_Polygon_getNumberOfSimplicesInDecomposition Interface Reference	244
6.309 f_polygon_class::F_Polygon_getPlaneOfExistence Interface Reference	244
6.310 f_polygon_class::F_Polygon_getPt Interface Reference	244
6.311 f_polygon_class::F_Polygon_getSimplexFromDecomposition Interface Reference	245
6.312 f_polygon_class::F_Polygon_new Interface Reference	245
6.313 f_polygon_class::F_Polygon_printToScreen Interface Reference	245
6.314 f_polygon_class::F_Polygon_reversePtOrdering Interface Reference	245
6.315 f_polygon_class::F_Polygon_setPlaneOfExistence Interface Reference	246
6.316 f_polygon_class::F_Polygon_zeroPolygon Interface Reference	246
6.317 f_polyhedron24_class::F_Polyhedron24_adjustCapToMatchVolume Interface Reference	246
6.318 f_polyhedron24_class::F_Polyhedron24_construct Interface Reference	246
6.319 f_polyhedron24_class::F_Polyhedron24_delete Interface Reference	247
6.320 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_adjustCapToMatchVolume Interface Reference	247
6.321 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_construct Interface Reference	247
6.322 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_delete Interface Reference	247

6.323 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getBoundingPts Interface Reference	248
6.324 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getData Interface Reference	248
6.325 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getPt Interface Reference	248
6.326 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_new Interface Reference	248
6.327 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setData Interface Reference	249
6.328 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setPt Interface Reference	249
6.329 f_polyhedron24_class::F_Polyhedron24_getBoundingPts Interface Reference	249
6.330 f_polyhedron24_class::F_Polyhedron24_getPt Interface Reference	249
6.331 f_polyhedron24_class::F_Polyhedron24_new Interface Reference	250
6.332 f_polyhedron24_class::F_Polyhedron24_setPt Interface Reference	250
6.333 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_add← Member Interface Reference	250
6.334 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_delete In- terface Reference	250
6.335 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_empty← Neighborhood Interface Reference	251
6.336 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_new In- terface Reference	251
6.337 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_set← CenterOfStencil Interface Reference	251
6.338 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_set← Member Interface Reference	251
6.339 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setSize Interface Reference	252
6.340 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_set← SurfaceArea Interface Reference	252
6.341 f_reconstructioninterface::F_reconstructionWithAdvectecdNormals_ListedVM_VMAN_RC Interface Reference	252
6.342 f_reconstructioninterface::F_reconstructionWithAdvectecdNormalsDebug_ListedVM_VMAN_RC In- terface Reference	252
6.343 f_reconstructioninterface::F_reconstructionWithELVIRA2D Interface Reference	253
6.344 f_reconstructioninterface::F_reconstructionWithELVIRA3D Interface Reference	253
6.345 f_reconstructioninterface::F_reconstructionWithLVIRA2D_RC Interface Reference	253
6.346 f_reconstructioninterface::F_reconstructionWithLVIRA3D_RC Interface Reference	253
6.347 f_reconstructioninterface::F_reconstructionWithMOF2D_RectangularCuboid Interface Reference	254

6.348 f_reconstructioninterface::F_reconstructionWithMOF2D_Tri Interface Reference	254
6.349 f_reconstructioninterface::F_reconstructionWithMOF2DGiveWeights_RectangularCuboid Interface Reference	254
6.350 f_reconstructioninterface::F_reconstructionWithMOF2DGiveWeights_Tri Interface Reference	254
6.351 f_reconstructioninterface::F_reconstructionWithMOF3D_RectangularCuboid Interface Reference	255
6.352 f_reconstructioninterface::F_reconstructionWithMOF3D_Tet Interface Reference	255
6.353 f_reconstructioninterface::F_reconstructionWithMOF3DGiveWeights_RectangularCuboid Interface Reference	255
6.354 f_reconstructioninterface::F_reconstructionWithMOF3DGiveWeights_Tet Interface Reference	255
6.355 f_reconstructioninterface::F_reconstructionWithR2P2D_RC Interface Reference	256
6.356 f_reconstructioninterface::F_reconstructionWithR2P2DDebug_RC Interface Reference	256
6.357 f_reconstructioninterface::F_reconstructionWithR2P3D_RC Interface Reference	256
6.358 f_reconstructioninterface::F_reconstructionWithR2P3DDebug_RC Interface Reference	256
6.359 f_rectangularcuboid_class::F_RectangularCuboid_calculateVolume Interface Reference	257
6.360 f_rectangularcuboid_class::F_RectangularCuboid_construct Interface Reference	257
6.361 f_rectangularcuboid_class::F_RectangularCuboid_construct_2pt Interface Reference	257
6.362 f_rectangularcuboid_class::F_RectangularCuboid_delete Interface Reference	257
6.363 f_rectangularcuboid_class::F_RectangularCuboid_getBoundingPts Interface Reference	258
6.364 f_rectangularcuboid_class::F_RectangularCuboid_new Interface Reference	258
6.365 f_sepvm_class::F_SepVM_construct Interface Reference	258
6.366 f_sepvm_class::F_SepVM_delete Interface Reference	258
6.367 f_sepvm_doubles3_class::F_SepVM_doubles3_delete Interface Reference	259
6.368 f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroid Interface Reference	259
6.369 f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroidPtr Interface Reference	259
6.370 f_sepvm_doubles3_class::F_SepVM_doubles3_getData Interface Reference	259
6.371 f_sepvm_doubles3_class::F_SepVM_doubles3_getVolume Interface Reference	260
6.372 f_sepvm_doubles3_class::F_SepVM_doubles3_getVolumePtr Interface Reference	260
6.373 f_sepvm_doubles3_class::F_SepVM_doubles3_multiplyByVolume Interface Reference	260
6.374 f_sepvm_doubles3_class::F_SepVM_doubles3_new Interface Reference	260
6.375 f_sepvm_doubles3_class::F_SepVM_doubles3_normalizeByVolume Interface Reference	261
6.376 f_sepvm_class::F_SepVM_getCentroid Interface Reference	261

6.377 f_sepvm_class::F_SepVM_getCentroidPtr Interface Reference	261
6.378 f_sepvm_class::F_SepVM_getVolume Interface Reference	261
6.379 f_sepvm_class::F_SepVM_getVolumePtr Interface Reference	262
6.380 f_sepvm_class::F_SepVM_multiplyByVolume Interface Reference	262
6.381 f_sepvm_class::F_SepVM_new Interface Reference	262
6.382 f_sepvm_class::F_SepVM_normalizeByVolume Interface Reference	262
6.383 f_serializer::F_Serializer_serializeAndPack_PlanarSeparator_ByteBuffer Interface Reference	263
6.384 f_serializer::F_Serializer_unpackAndStore_PlanarSeparator_ByteBuffer Interface Reference	263
6.385 f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS Interface Reference	263
6.386 f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS_DefTol Interface Reference	263
6.387 f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_append Interface Reference	264
6.388 f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_clear Interface Reference	264
6.389 f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_delete Interface Reference	264
6.390 f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getListAtIndex Interface Reference	264
6.391 f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getSize Interface Reference	265
6.392 f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getTagForIndex Interface Reference	265
6.393 f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_new Interface Reference	265
6.394 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_delete Interface Reference	265
6.395 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtIndex Interface Reference	266
6.396 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtTag Interface Reference	266
6.397 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidPtrAtIndex Interface Reference	266
6.398 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getSize Interface Reference	266
6.399 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getTagForIndex Interface Reference	267
6.400 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtIndex Interface Reference	267

6.401 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtTag Interface Reference	267
6.402 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumePtrAtIndex Interface Reference	267
6.403 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_multiplyByVolume Interface Reference	268
6.404 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_new Interface Reference	268
6.405 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_normalizeByVolume Interface Reference	268
6.406 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_delete Interface Reference	268
6.407 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidAtIndex Interface Reference	269
6.408 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidPtrAtIndex Interface Reference	269
6.409 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getSize Interface Reference	269
6.410 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getTagForIndex Interface Reference	269
6.411 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumeAtIndex Interface Reference	270
6.412 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumePtrAtIndex Interface Reference	270
6.413 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_multiplyByVolume Interface Reference	270
6.414 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_new Interface Reference	270
6.415 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_normalizeByVolume Interface Reference	271
6.416 f_tet_class::F_Tet_construct Interface Reference	271
6.417 f_tet_class::F_Tet_delete Interface Reference	271
6.418 f_tet_class::F_Tet_getBoundingPts Interface Reference	271
6.419 f_tet_class::F_Tet_new Interface Reference	272
6.420 f_tri_class::F_Tri_calculateAndSetPlaneOfExistence Interface Reference	272
6.421 f_tri_class::F_Tri_calculateCentroid Interface Reference	272
6.422 f_tri_class::F_Tri_calculateNormal Interface Reference	272
6.423 f_tri_class::F_Tri_calculateSign Interface Reference	273
6.424 f_tri_class::F_Tri_calculateVolume Interface Reference	273
6.425 f_tri_class::F_Tri_construct Interface Reference	273
6.426 f_tri_class::F_Tri_delete Interface Reference	273
6.427 f_tri_class::F_Tri_getBoundingPts Interface Reference	274
6.428 f_tri_class::F_Tri_getLocalizer Interface Reference	274

6.429 f_tri_class::F_Tri_getPlaneOfExistence Interface Reference	274
6.430 f_tri_class::F_Tri_getVertices Interface Reference	274
6.431 f_tri_class::F_Tri_new Interface Reference	275
6.432 f_tri_class::F_Tri_reversePtOrdering Interface Reference	275
6.433 f_tri_class::F_Tri_setPlaneOfExistence Interface Reference	275
6.434 f_vman_class::F_VMAN_delete Interface Reference	275
6.435 f_vman_class::F_VMAN_getCentroid Interface Reference	276
6.436 f_vman_class::F_VMAN_getNormal Interface Reference	276
6.437 f_vman_class::F_VMAN_getVolume Interface Reference	276
6.438 f_vman_class::F_VMAN_multiplyByVolume Interface Reference	276
6.439 f_vman_class::F_VMAN_new Interface Reference	277
6.440 f_vman_class::F_VMAN_normalizeByVolume Interface Reference	277
6.441 IRL::Face< HalfEdgeType > Class Template Reference	277
6.442 IRL::GeneralPolygon< Derived, VertexType, SimplexType > Class Template Reference	279
6.443 IRL::GeneralPolyhedron< Derived, VertexType, SimplexType > Class Template Reference	281
6.444 f_cappeddodecahedron_class::getboundingpts Interface Reference	283
6.445 f_tet_class::getboundingpts Interface Reference	283
6.446 f_tri_class::getboundingpts Interface Reference	284
6.447 f_dodecahedron_class::getboundingpts Interface Reference	284
6.448 f_cappeddodecahedron_doubles3_class::getboundingpts Interface Reference	284
6.449 f_polygon_class::getboundingpts Interface Reference	284
6.450 f_polyhedron24_class::getboundingpts Interface Reference	285
6.451 f_polyhedron24_doubles3_class::getboundingpts Interface Reference	285
6.452 f_rectangularcuboid_class::getboundingpts Interface Reference	285
6.453 f_dividedpolygon_class::getboundingpts Interface Reference	285
6.454 f_vman_class::getcentroid Interface Reference	286
6.455 f_sepvm_class::getcentroid Interface Reference	286
6.456 f_sepvm_doubles3_class::getcentroid Interface Reference	286
6.457 f_tagged_accumvm_sepvm_class::getcentroidatindex Interface Reference	286
6.458 f_tagged_accumvm_vm_class::getcentroidatindex Interface Reference	287

6.459 f_tagged_accumvm_sepvm_class::getcentroidattag Interface Reference	287
6.460 f_sepvm_class::getcentroidptr Interface Reference	287
6.461 f_sepvm_doubles3_class::getcentroidptr Interface Reference	287
6.462 f_tagged_accumvm_vm_class::getcentroidptratindex Interface Reference	288
6.463 f_tagged_accumvm_sepvm_class::getcentroidptratindex Interface Reference	288
6.464 f_objectallocationserver_localizedseparatorlink_class::getcobject Interface Reference	288
6.465 f_tagged_accumvm_sepvm_class::getcobject Interface Reference	288
6.466 f_objectallocationserver_localizerlink_class::getcobject Interface Reference	289
6.467 f_objectallocationserver_planarlocalizer_class::getcobject Interface Reference	289
6.468 f_tagged_accumvm_vm_class::getcobject Interface Reference	289
6.469 f_objectallocationserver_planarseparator_class::getcobject Interface Reference	289
6.470 f_planarlocalizer_class::getcobject Interface Reference	290
6.471 f_tet_class::getcobject Interface Reference	290
6.472 f_tri_class::getcobject Interface Reference	290
6.473 f_planarseparator_class::getcobject Interface Reference	290
6.474 f_dodecahedron_class::getcobject Interface Reference	291
6.475 f_vman_class::getcobject Interface Reference	291
6.476 f_cappeddodecahedron_doubles3_class::getcobject Interface Reference	291
6.477 f_elviraneighborhood_class::getcobject Interface Reference	291
6.478 f_polygon_class::getcobject Interface Reference	292
6.479 f_polyhedron24_class::getcobject Interface Reference	292
6.480 f_polyhedron24_doubles3_class::getcobject Interface Reference	292
6.481 f_listedvm_vman_class::getcobject Interface Reference	292
6.482 f_bytebuffer_class::getcobject Interface Reference	293
6.483 f_r2pneighborhood_rectangularcuboid_class::getcobject Interface Reference	293
6.484 f_localizedseparatorlink_class::getcobject Interface Reference	293
6.485 f_dividedpolygon_class::getcobject Interface Reference	293
6.486 f_rectangularcuboid_class::getcobject Interface Reference	294
6.487 f_localizerlink_class::getcobject Interface Reference	294
6.488 f_sepvm_class::getcobject Interface Reference	294

6.489 f_sepvm_doubles3_class::getcobject Interface Reference	294
6.490 f_lviraneighborhood_rectangularcuboid_class::getcobject Interface Reference	295
6.491 f_tagged_accumlistedvm_vman_class::getcobject Interface Reference	295
6.492 f_cappeddodecahedron_class::getcobject Interface Reference	295
6.493 f_cappeddodecahedron_doubles3_class::getdata Interface Reference	295
6.494 f_polyhedron24_doubles3_class::getdata Interface Reference	296
6.495 f_sepvm_doubles3_class::getdata Interface Reference	296
6.496 f_localizedseparatorlink_class::getid Interface Reference	296
6.497 f_localizerlink_class::getid Interface Reference	296
6.498 f_tagged_accumlistedvm_vman_class::getlistatindex Interface Reference	297
6.499 f_tri_class::getlocalizer Interface Reference	297
6.500 f_polygon_class::getlocalizer Interface Reference	297
6.501 f_dividedpolygon_class::getlocalizer Interface Reference	297
6.502 f_listedvm_vman_class::getmoments Interface Reference	298
6.503 f_vman_class::getnormal Interface Reference	298
6.504 f_getvolumemoments::getnormalizedvolumemoments Interface Reference	298
6.505 f_planarseparator_class::getnumberofplanes Interface Reference	299
6.506 f_polygon_class::getnumberofsimplicesindecomposition Interface Reference	299
6.507 f_dividedpolygon_class::getnumberofsimplicesindecomposition Interface Reference	299
6.508 f_polygon_class::getnumberofvertices Interface Reference	299
6.509 f_dividedpolygon_class::getnumberofvertices Interface Reference	300
6.510 f_planarseparator_class::getplane Interface Reference	300
6.511 f_tri_class::getplaneofexistence Interface Reference	300
6.512 f_polygon_class::getplaneofexistence Interface Reference	300
6.513 f_dividedpolygon_class::getplaneofexistence Interface Reference	301
6.514 f_cutpolygon::getplanepolygonfromreconstruction Interface Reference	301
6.515 f_cappeddodecahedron_class::getpt Interface Reference	301
6.516 f_cappeddodecahedron_doubles3_class::getpt Interface Reference	301
6.517 f_polygon_class::getpt Interface Reference	302
6.518 f_polyhedron24_class::getpt Interface Reference	302

6.519 f_polyhedron24_doubles3_class::getpt Interface Reference	302
6.520 f_dividedpolygon_class::getpt Interface Reference	302
6.521 f_cutpolygon::getreconstructionsurfacearea Interface Reference	303
6.522 f_polygon_class::getsimplexfromdecomposition Interface Reference	303
6.523 f_dividedpolygon_class::getsimplexfromdecomposition Interface Reference	303
6.524 f_tagged_accumvm_vm_class::getsize Interface Reference	303
6.525 f_tagged_accumvm_sepvm_class::getsize Interface Reference	304
6.526 f_listedvm_vman_class::getsize Interface Reference	304
6.527 f_bytebuffer_class::getsize Interface Reference	304
6.528 f_tagged_accumlistedvm_vman_class::getsize Interface Reference	304
6.529 f_tagged_accumvm_sepvm_class::gettagforindex Interface Reference	305
6.530 f_tagged_accumvm_vm_class::gettagforindex Interface Reference	305
6.531 f_tagged_accumlistedvm_vman_class::gettagforindex Interface Reference	305
6.532 f_tri_class::getvertices Interface Reference	305
6.533 f_vman_class::getvolume Interface Reference	306
6.534 f_sepvm_class::getvolume Interface Reference	306
6.535 f_sepvm_doubles3_class::getvolume Interface Reference	306
6.536 f_tagged_accumvm_sepvm_class::getvolumeatindex Interface Reference	306
6.537 f_tagged_accumvm_vm_class::getvolumeatindex Interface Reference	307
6.538 f_tagged_accumvm_sepvm_class::getvolumeattag Interface Reference	307
6.539 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, Encompassing< Type, ReconstructionType, Enable > Struct Template Reference	307
6.539.1 Detailed Description	307
6.540 f_getvolumemoments::getvolumemoments Interface Reference	308
6.541 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, Encompassing< Type, NullReconstruction, enable_if_t< IsNullReconstruction< NullReconstruction >::value > > Struct Template Reference	308
6.542 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, Encompassing< Type, PlanarSeparator, enable_if_t< is_separated_moments< ReturnType >::value > > Struct Template Reference	309
6.543 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, Encompassing< Type, ReconstructionType, enable_if_t< isHalfEdgeCutting< CuttingMethod >::value && IsNot< ANullReconstruction< ReconstructionType >::value && !(IsPlanarSeparator< ReconstructionType >::value && is_separated_moments< ReturnType >::value) > > Struct Template Reference	310

6.544 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isRecursiveSimplexCutting< CuttingMethod >::value &&!IsNotANullReconstruction< ReconstructionType >::value &&!IsPlanarSeparator< ReconstructionType >::value &&is_separated_moments< ReturnType >::value> > Struct Template Reference	310
6.545 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isSimplexCutting< CuttingMethod >::value &&IsNotANullReconstruction< ReconstructionType >::value &&!IsPlanarSeparator< ReconstructionType >::value &&is_separated_moments< ReturnType >::value> > Struct Template Reference	311
6.546 f_getvolumemoments::getvolumemoments_setmethod Interface Reference	311
6.547 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, ReconstructionType, Enable > Struct Template Reference	311
6.548 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, NullReconstruction, enable_if_t< IsNull< Reconstruction< NullReconstruction >::value > > Struct Template Reference	312
6.549 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, PlanarSeparator, enable_if_t< isHalfEdge< Cutting< CuttingMethod >::value &&is_separated_moments< ReturnType >::value > > Struct Template Reference	312
6.550 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, PlanarSeparator, enable_if_t< isSimplexCutting< CuttingMethod >::value &&is_separated_moments< ReturnType >::value > > Struct Template Reference	314
6.551 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, ReconstructionType, enable_if_t< isHalfEdge< Cutting< CuttingMethod >::value &&!IsNotANullReconstruction< ReconstructionType >::value &&!IsPlanarSeparator< ReconstructionType >::value &&is_separated_moments< ReturnType >::value> > Struct Template Reference	315
6.552 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, ReconstructionType, enable_if_t< isSimplexCutting< CuttingMethod >::value &&!IsNotANullReconstruction< ReconstructionType >::value &&!IsPlanarSeparator< ReconstructionType >::value &&is_separated_moments< ReturnType >::value> > Struct Template Reference	315
6.553 f_sepvm_class::getvolumeptr Interface Reference	316
6.554 f_sepvm_doubles3_class::getvolumeptr Interface Reference	316
6.555 f_tagged_accumvm_vm_class::getvolumepratindex Interface Reference	316
6.556 f_tagged_accumvm_sepvm_class::getvolumepratindex Interface Reference	316
6.557 IRL::HalfEdge< VertexType > Class Template Reference	317
6.558 IRL::HalfEdgeCutting Struct Reference	318
6.559 IRL::HalfEdgePolygon< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces > Class Template Reference	318

6.560 IRL::HalfEdgePolyhedron< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, k← MaxHalfEdges, kMaxVertices, kMaxFaces > Class Template Reference	320
6.561 IRL::HalfEdgePolytope< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, k← MaxHalfEdges, kMaxVertices, kMaxFaces > Class Template Reference	321
6.562 IRL::has_localizer< C > Struct Template Reference	322
6.563 IRL::has_localizer< const C > Struct Template Reference	323
6.564 IRL::has_localizer< LocalizedSeparator > Struct Template Reference	324
6.565 IRL::has_localizer< LocalizedSeparatorGroup > Struct Template Reference	325
6.566 IRL::has_localizer< LocalizedSeparatorGroupLink > Struct Template Reference	326
6.567 IRL::has_localizer< LocalizedSeparatorLink > Struct Template Reference	327
6.568 IRL::has_localizer< LocalizerLink > Struct Template Reference	328
6.569 IRL::has_localizer< LocalizerLinkFromLocalizedSeparatorLink > Struct Template Reference	329
6.570 IRL::has_localizer< MaskedLocalizedSeparatorLink > Struct Template Reference	330
6.571 IRL::has_localizer< PlanarLocalizer > Struct Template Reference	331
6.572 IRL::has_localizer< RestrictedLocalizerLinkFromLocalizedSeparatorLink > Struct Template Refer- ence	332
6.573 IRL::has_localizer< RestrictedMaskedLocalizedSeparatorLink > Struct Template Reference	333
6.574 IRL::has_null_reconstruction< C > Struct Template Reference	334
6.575 IRL::has_null_reconstruction< const C > Struct Template Reference	335
6.576 IRL::has_null_reconstruction< NullReconstruction > Struct Template Reference	336
6.577 IRL::has_separator< C > Struct Template Reference	337
6.578 IRL::has_separator< const C > Struct Template Reference	338
6.579 IRL::has_separator< LocalizedSeparator > Struct Template Reference	339
6.580 IRL::has_separator< LocalizedSeparatorGroup > Struct Template Reference	340
6.581 IRL::has_separator< LocalizedSeparatorGroupLink > Struct Template Reference	341
6.582 IRL::has_separator< LocalizedSeparatorLink > Struct Template Reference	342
6.583 IRL::has_separator< MaskedLocalizedSeparatorLink > Struct Template Reference	343
6.584 IRL::has_separator< PlanarSeparator > Struct Template Reference	344
6.585 IRL::has_separator< PlanarSeparatorPath > Struct Template Reference	345
6.586 IRL::has_separator< RestrictedMaskedLocalizedSeparatorLink > Struct Template Reference	346
6.587 IRL::HasACollection< ReturnType > Struct Template Reference	347
6.588 IRL::HasALocalizer< ReconstructionType > Struct Template Reference	347

6.589 IRL::HasANestedType< ReturnType > Struct Template Reference	348
6.590 IRL::HasAReconstructionLink< ReconstructionType > Struct Template Reference	348
6.590.1 Member Data Documentation	348
6.590.1.1 value	348
6.591 IRL::HasASeparator< ReconstructionType > Struct Template Reference	348
6.592 IRL::HasLocalizer_AND_HasSeparator< ReconstructionType > Struct Template Reference	349
6.592.1 Member Data Documentation	349
6.592.1.1 value	349
6.593 IRL::HasLocalizer_OR_HasSeparator< ReconstructionType > Struct Template Reference	349
6.593.1 Member Data Documentation	349
6.593.1.1 value	349
6.594 IRL::HasLocalizer_XOR_HasSeparator< ReconstructionType > Struct Template Reference	350
6.594.1 Member Data Documentation	350
6.594.1.1 value	350
6.595 IRL::HexahedronSpecialization< Derived, VertexType > Class Template Reference	350
6.595.1 Detailed Description	351
6.596 IRL::IgnoreVolumeAbovePlane Struct Reference	351
6.597 IRL::Illinois< OptimizingClass > Class Template Reference	351
6.597.1 Detailed Description	352
6.597.2 Member Function Documentation	352
6.597.2.1 solve() [1/2]	352
6.597.2.2 solve() [2/2]	353
6.598 IRL::is_moments_collection< C > Struct Template Reference	353
6.598.1 Detailed Description	354
6.599 IRL::is_moments_collection< AccumulatedListedVolumeMoments< VolumeMomentsType > > Struct Template Reference	355
6.599.1 Detailed Description	355
6.600 IRL::is_moments_collection< AccumulatedVolumeMoments< VolumeMomentsType > > Struct Template Reference	356
6.600.1 Detailed Description	356
6.601 IRL::is_moments_collection< const C > Struct Template Reference	357

6.602 IRL::is_moments_collection< ListedVolumeMoments< VolumeMomentsType > > Struct Template Reference	358
6.602.1 Detailed Description	358
6.603 IRL::is_moments_collection< TaggedAccumulatedListedVolumeMoments< VolumeMomentsType > > Struct Template Reference	359
6.603.1 Detailed Description	359
6.604 IRL::is_moments_collection< TaggedAccumulatedVolumeMoments< VolumeMomentsType > > Struct Template Reference	360
6.605 IRL::is_nested_moments< C > Struct Template Reference	360
6.605.1 Detailed Description	361
6.606 IRL::is_nested_moments< AccumulateWrapper< NestedType > > Struct Template Reference	362
6.606.1 Detailed Description	362
6.607 IRL::is_nested_moments< const C > Struct Template Reference	363
6.608 IRL::is_planar_separator< C > Struct Template Reference	364
6.609 IRL::is_planar_separator< const C > Struct Template Reference	365
6.610 IRL::is_planar_separator< PlanarSeparator > Struct Template Reference	366
6.611 IRL::is_polygon< C > Struct Template Reference	367
6.612 IRL::is_polygon< const C > Struct Template Reference	368
6.613 IRL::is_polygon< ExpandableDividedPolygon< VertexType > > Struct Template Reference	369
6.614 IRL::is_polygon< ExpandablePolygon< VertexType > > Struct Template Reference	370
6.615 IRL::is_polygon< ProxyTri< GeometryType > > Struct Template Reference	371
6.616 IRL::is_polygon< SegmentedDecomposedPolygon< VertexType > > Struct Template Reference	372
6.617 IRL::is_polygon< SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices > > Struct Template Reference	373
6.618 IRL::is_polygon< StoredTri< VertexType > > Struct Template Reference	374
6.619 IRL::is_polyhedron< C > Struct Template Reference	375
6.620 IRL::is_polyhedron< const C > Struct Template Reference	376
6.621 IRL::is_polyhedron< ProxyTet< GeometryType > > Struct Template Reference	377
6.622 IRL::is_polyhedron< SegmentedDecomposedPolyhedron< VertexType > > Struct Template Reference	378
6.623 IRL::is_polyhedron< SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices > > Struct Template Reference	379
6.624 IRL::is_polyhedron< StoredCappedDodecahedron< VertexType > > Struct Template Reference	380

6.625 IRL::is_polyhedron< StoredConcaveBox< VertexType > > Struct Template Reference	381
6.626 IRL::is_polyhedron< StoredDodecahedron< VertexType > > Struct Template Reference	382
6.627 IRL::is_polyhedron< StoredHexahedron< VertexType > > Struct Template Reference	383
6.628 IRL::is_polyhedron< StoredPolyhedron24< VertexType > > Struct Template Reference	384
6.629 IRL::is_polyhedron< StoredRectangularCuboid< VertexType > > Struct Template Reference	385
6.630 IRL::is_polyhedron< StoredTet< VertexType > > Struct Template Reference	386
6.631 IRL::is_reconstruction_link< C > Struct Template Reference	387
6.632 IRL::is_reconstruction_link< const C > Struct Template Reference	388
6.633 IRL::is_reconstruction_link< LocalizedSeparatorGroupLink > Struct Template Reference	389
6.634 IRL::is_reconstruction_link< LocalizedSeparatorLink > Struct Template Reference	390
6.635 IRL::is_reconstruction_link< LocalizerLink > Struct Template Reference	391
6.636 IRL::is_reconstruction_link< LocalizerLinkFromLocalizedSeparatorLink > Struct Template Reference	392
6.637 IRL::is_reconstruction_link< MaskedLocalizedSeparatorLink > Struct Template Reference	393
6.638 IRL::is_reconstruction_link< PlanarSeparatorPath > Struct Template Reference	394
6.639 IRL::is_reconstruction_link< RestrictedLocalizerLinkFromLocalizedSeparatorLink > Struct Template Reference	395
6.640 IRL::is_reconstruction_link< RestrictedMaskedLocalizedSeparatorLink > Struct Template Reference	396
6.641 IRL::is_separated_moments< C > Struct Template Reference	396
6.641.1 Detailed Description	397
6.642 IRL::is_separated_moments< const C > Struct Template Reference	398
6.643 IRL::is_separated_moments< SeparatedMoments< MomentsType > > Struct Template Reference	399
6.644 IRL::is_tet< C > Struct Template Reference	400
6.645 IRL::is_tet< const C > Struct Template Reference	401
6.646 IRL::is_tet< ProxyTet< GeometryType > > Struct Template Reference	402
6.647 IRL::is_tet< StoredTet< VertexType > > Struct Template Reference	403
6.648 IRL::is_tri< C > Struct Template Reference	404
6.649 IRL::is_tri< const C > Struct Template Reference	405
6.650 IRL::is_tri< ProxyTri< GeometryType > > Struct Template Reference	406
6.651 IRL::is_tri< StoredTri< VertexType > > Struct Template Reference	407
6.652 f_planarseparator_class::isflipped Interface Reference	407
6.653 IRL::isHalfEdgeCutting< C > Struct Template Reference	408

6.654 IRL::isHalfEdgeCutting< const C > Struct Template Reference	409
6.655 IRL::isHalfEdgeCutting< HalfEdgeCutting > Struct Template Reference	410
6.656 IRL::IsNotANullReconstruction< ReconstructionType > Struct Template Reference	410
6.656.1 Member Data Documentation	410
6.656.1.1 value	411
6.657 IRL::IsNotAPlanarSeparator< ReconstructionType > Struct Template Reference	411
6.658 IRL::IsNullReconstruction< ReconstructionType > Struct Template Reference	411
6.658.1 Member Data Documentation	411
6.658.1.1 value	411
6.659 IRL::IsPlanarSeparator< ReconstructionType > Struct Template Reference	412
6.660 f_geometriccuttinghelpers::isptinternal Interface Reference	412
6.661 IRL::isRecursiveSimplexCutting< C > Struct Template Reference	412
6.662 IRL::isRecursiveSimplexCutting< const C > Struct Template Reference	413
6.663 IRL::isRecursiveSimplexCutting< RecursiveSimplexCutting > Struct Template Reference	414
6.664 IRL::isSimplexCutting< C > Struct Template Reference	415
6.665 IRL::isSimplexCutting< const C > Struct Template Reference	416
6.666 IRL::isSimplexCutting< SimplexCutting > Struct Template Reference	417
6.667 IRL::IterativeSolverForDistance< CellType, kMaxPlanes > Class Template Reference	418
6.667.1 Detailed Description	420
6.667.2 Constructor & Destructor Documentation	420
6.667.2.1 IterativeSolverForDistance() [1/2]	420
6.667.2.2 IterativeSolverForDistance() [2/2]	421
6.667.3 Member Function Documentation	421
6.667.3.1 solve()	421
6.668 IRL::IteratorThroughBracketOperator< ContainerType > Class Template Reference	422
6.669 IRL::JoinedReconstructions< CurrentReconstruction, NextReconstruction > Class Template Reference	423
6.669.1 Detailed Description	424
6.670 IRL::JoinedReconstructions< CurrentReconstruction, NullReconstruction > Class Template Reference	424
6.671 IRL::JoinedReconstructionsToGroup< CurrentReconstruction, GroupReconstruction > Class Template Reference	425

6.672 IRL::KMeans Struct Reference	425
6.672.1 Detailed Description	426
6.673 IRL::LevenbergMarquardt< OptimizingClass, kRows, kColumns > Class Template Reference	426
6.673.1 Detailed Description	427
6.673.2 Member Data Documentation	428
6.673.2.1 reason_for_exit_m	428
6.674 IRL::LevenbergMarquardt< OptimizingClass, -1, kColumns > Class Template Reference	429
6.674.1 Member Data Documentation	430
6.674.1.1 reason_for_exit_m	430
6.675 IRL::LinearInterpolation_Functor Class Reference	430
6.676 f_listedvdm_vman_class::listedvdm_vman_type Type Reference	430
6.677 IRL::ListedVolumeMoments< VolumeMomentsType > Class Template Reference	431
6.677.1 Detailed Description	432
6.678 IRL::Lister< ObjectType > Class Template Reference	433
6.678.1 Detailed Description	434
6.679 f_localizedseparatorlink_class::localizedseparatorlink_type Type Reference	434
6.680 f_localizerlink_class::localizerlink_type Type Reference	435
6.681 IRL::LocalizerLinkFromLocalizedSeparatorLink Class Reference	435
6.682 IRL::LVIRA_2D< CellType > Class Template Reference	436
6.682.1 Detailed Description	438
6.682.2 Member Function Documentation	438
6.682.2.1 getDeltaRotationQuat()	438
6.683 IRL::LVIRA_3D< CellType > Class Template Reference	439
6.683.1 Detailed Description	441
6.683.2 Member Function Documentation	441
6.683.2.1 getDeltaRotationQuat()	441
6.684 IRL::LVIRACommon< CellType, kColumns > Class Template Reference	442
6.684.1 Detailed Description	444
6.684.2 Member Function Documentation	444
6.684.2.1 runOptimization()	445

6.684.3 Member Data Documentation	445
6.684.3.1 fininite_difference_angle_m	445
6.685 IRL::LVIRADebug< LVIRAType > Class Template Reference	446
6.685.1 Detailed Description	447
6.686 IRL::LVIRANeighborhood< CellType > Class Template Reference	447
6.686.1 Detailed Description	449
6.687 f_lviraneighborhood_rectangularcuboid_class::lviraneighborhood_rectangularcuboid_type Type Reference	450
6.688 IRL::MaskedLocalizedSeparatorLink Class Reference	450
6.689 IRL::MaskStripper< MaskedType > Class Template Reference	451
6.690 IRL::MathVector< kNumberOfElements > Class Template Reference	452
6.691 IRL::MOF_2D< CellType > Class Template Reference	453
6.692 IRL::MOF_3D< CellType > Class Template Reference	455
6.693 IRL::MOFCommon< CellType > Class Template Reference	456
6.693.1 Member Data Documentation	458
6.693.1.1 initial_angle_m	459
6.694 IRL::MOFDebug< MOFType > Class Template Reference	459
6.694.1 Detailed Description	460
6.695 IRL::MomentCalculationType< VertexType, ProxyType, Enable > Class Template Reference	461
6.696 IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polygon< ProxyType >::value > > Class Template Reference	461
6.697 IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polyhedron< ProxyType >::value > > Class Template Reference	461
6.698 IRL::MultiMaterialAccumulatedVolumeMoments< MomentsType > Class Template Reference	462
6.698.1 Detailed Description	463
6.699 f_tagged_accumvm_sepvm_class::multiplybyvolume Interface Reference	463
6.700 f_tagged_accumvm_vm_class::multiplybyvolume Interface Reference	464
6.701 f_vman_class::multiplybyvolume Interface Reference	464
6.702 f_sepvm_class::multiplybyvolume Interface Reference	464
6.703 f_sepvm_doubles3_class::multiplybyvolume Interface Reference	464
6.704 f_objectallocationserver_localizedseparatorlink_class::new Interface Reference	465
6.705 f_elviraneighborhood_class::new Interface Reference	465

6.706 f_objectallocationserver_planarlocalizer_class::new Interface Reference	465
6.707 f_planarlocalizer_class::new Interface Reference	465
6.708 f_tet_class::new Interface Reference	466
6.709 f_polygon_class::new Interface Reference	466
6.710 f_dodecahedron_class::new Interface Reference	466
6.711 f_planarseparator_class::new Interface Reference	466
6.712 f_rectangularcuboid_class::new Interface Reference	467
6.713 f_vman_class::new Interface Reference	467
6.714 f_tri_class::new Interface Reference	467
6.715 f_cappeddodecahedron_doubles3_class::new Interface Reference	467
6.716 f_tagged_accumvm_vm_class::new Interface Reference	468
6.717 f_tagged_accumlistedvm_vman_class::new Interface Reference	468
6.718 f_dividedpolygon_class::new Interface Reference	468
6.719 f_polyhedron24_class::new Interface Reference	468
6.720 f_polyhedron24_doubles3_class::new Interface Reference	469
6.721 f_listedvm_vman_class::new Interface Reference	469
6.722 f_bytebuffer_class::new Interface Reference	469
6.723 f_objectallocationserver_planarseparator_class::new Interface Reference	469
6.724 f_r2pnighborhood_rectangularcuboid_class::new Interface Reference	470
6.725 f_localizedseparatorlink_class::new Interface Reference	470
6.726 f_localizerlink_class::new Interface Reference	470
6.727 f_objectallocationserver_localizerlink_class::new Interface Reference	470
6.728 f_sepvm_class::new Interface Reference	471
6.729 f_sepvm_doubles3_class::new Interface Reference	471
6.730 f_lviraneighborhood_rectangularcuboid_class::new Interface Reference	471
6.731 f_tagged_accumvm_sepvm_class::new Interface Reference	471
6.732 f_cappeddodecahedron_class::new Interface Reference	472
6.733 IRL::Normal Class Reference	472
6.733.1 Detailed Description	474
6.734 f_tagged_accumvm_sepvm_class::normalizebyvolume Interface Reference	474

6.735 f_tagged_accumvm_vm_class::normalizebyvolume Interface Reference	474
6.736 f_sepvm_class::normalizebyvolume Interface Reference	475
6.737 f_sepvm_doubles3_class::normalizebyvolume Interface Reference	475
6.738 f_vman_class::normalizebyvolume Interface Reference	475
6.739 IRL::NullReconstruction Class Reference	475
6.739.1 Detailed Description	475
6.740 IRL::ObjectAllocationServer< ObjectType > Class Template Reference	476
6.741 f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_type Type Reference	476
6.742 f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_type Type Reference	477
6.743 f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer_type Type Reference	477
6.744 f_objectallocationserver_planarseparator_class::objectallocationserver_planarseparator_type Type Reference	478
6.745 IRL::PartitionByNormal< MomentsContainerType > Class Template Reference	478
6.745.1 Detailed Description	480
6.746 IRL::PassToNestedType Struct Reference	480
6.747 IRL::PathGraphNode< NodeType > Class Template Reference	480
6.747.1 Detailed Description	481
6.748 IRL::PlanarLocalizer Class Reference	482
6.748.1 Detailed Description	484
6.749 f_planarlocalizer_class::planarlocalizer_type Type Reference	484
6.750 IRL::PlanarReconstruction< kStackPlanes > Class Template Reference	485
6.751 IRL::PlanarSeparator Class Reference	487
6.752 f_planarseparator_class::planarseparator_type Type Reference	489
6.753 IRL::PlanarSeparatorPathGroup Class Reference	490
6.753.1 Detailed Description	490
6.754 IRL::Plane Class Reference	491
6.754.1 Detailed Description	492
6.755 f_polygon_class::polygon_type Type Reference	492
6.756 IRL::PolygonBase< VertexType > Class Template Reference	493

6.757 IRL::PolygonConstructors< Derived, VertexType, StorageType, PlaneOfExistenceType > Class Template Reference	493
6.758 IRL::PolygonMomentsCalculation< Derived, VertexType, SimplexType > Class Template Reference	494
6.759 IRL::PolygonMomentsCalculationCommon< Derived, VertexType, SimplexType > Class Template Reference	495
6.760 IRL::PolygonSpecialization< Derived, VertexType > Class Template Reference	496
6.760.1 Detailed Description	498
6.761 IRL::Polyhedron< VertexType, kMaxNumberOfPts > Class Template Reference	499
6.762 f_polyhedron24_doubles3_class::polyhedron24_doubles3_type Type Reference	499
6.763 f_polyhedron24_class::polyhedron24_type Type Reference	500
6.764 IRL::Polyhedron24Specialization< Derived, VertexType > Class Template Reference	500
6.765 IRL::PolyhedronMomentsCalculation< Derived, VertexType, SimplexType > Class Template Reference	501
6.766 IRL::PolyhedronMomentsCalculation< Derived, PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, SimplexType > Class Template Reference	502
6.767 IRL::PolyhedronMomentsCalculationCommon< Derived, VertexType, SimplexType > Class Template Reference	503
6.768 f_dividedpolygon_class::printtoscreen Interface Reference	504
6.769 f_planarlocalizer_class::printtoscreen Interface Reference	504
6.770 f_planarseparator_class::printtoscreen Interface Reference	504
6.771 f_polygon_class::printtoscreen Interface Reference	505
6.772 IRL::ProxyPolygonConstructors< Derived, GeometryType, kNumberOfVertices, PlaneOfExistenceType > Class Template Reference	505
6.773 IRL::ProxyTet< GeometryType > Class Template Reference	506
6.773.1 Detailed Description	507
6.774 IRL::ProxyTri< GeometryType > Class Template Reference	507
6.775 IRL::ProxyVertexAccess< Derived, GeometryType, kNumberOfVertices > Class Template Reference	508
6.776 IRL::Pt Class Reference	509
6.776.1 Detailed Description	512
6.776.2 Member Function Documentation	512
6.776.2.1 fromEdgeIntersection()	512
6.777 IRL::PtList< VertexType, kMaxNumberOfPts > Class Template Reference	513
6.778 IRL::PtWithDataCommon< Derived, AttachedDataType > Class Template Reference	514

6.779 IRL::PtWithDoublesStatelessFunctor< FunctorType, kArrayLength > Class Template Reference	515
6.780 IRL::R2P_2D1P< CellType > Class Template Reference	516
6.780.1 Detailed Description	518
6.780.2 Member Function Documentation	518
6.780.2.1 getDeltaRotationQuat()	518
6.780.2.2 updateGuess()	518
6.781 IRL::R2P_2D2P< CellType > Class Template Reference	519
6.781.1 Detailed Description	520
6.781.2 Member Function Documentation	521
6.781.2.1 getDeltaRotationQuat()	521
6.781.2.2 updateGuess()	521
6.782 IRL::R2P_3D1P< CellType > Class Template Reference	521
6.782.1 Detailed Description	523
6.782.2 Member Function Documentation	523
6.782.2.1 getDeltaRotationQuat()	523
6.782.2.2 updateGuess()	524
6.783 IRL::R2P_3D2P< CellType > Class Template Reference	524
6.783.1 Detailed Description	526
6.783.2 Member Function Documentation	526
6.783.2.1 getDeltaRotationQuat()	526
6.783.2.2 updateGuess()	526
6.784 IRL::R2PCommon< CellType, kColumns > Class Template Reference	527
6.784.1 Detailed Description	529
6.784.2 Member Function Documentation	530
6.784.2.1 addCellMomentsToGeometryVector()	530
6.784.2.2 fillGeometryAndWeightVectors()	530
6.784.2.3 runOptimization()	531
6.784.2.4 setRelativeImportanceBetweenWeights()	531
6.784.3 Member Data Documentation	532
6.784.3.1 initial_angle_m	532

6.785 IRL::R2PDebug< R2PType > Class Template Reference	532
6.785.1 Detailed Description	534
6.786 IRL::R2PNeighborhood< CellType > Class Template Reference	534
6.786.1 Detailed Description	536
6.787 f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_type Type Reference	536
6.788 short_alloc::short_alloc< T, N, Align >::rebind< _Up > Struct Template Reference	537
6.789 IRL::ReconstructionLink< ReconstructionType, GraphType > Class Template Reference	537
6.789.1 Detailed Description	538
6.790 f_reconstructioninterface::reconstructionwithadvectedenormals Interface Reference	538
6.791 f_reconstructioninterface::reconstructionwithadvectedenormalsdebug Interface Reference	539
6.792 f_reconstructioninterface::reconstructionwithlvira2d Interface Reference	539
6.793 f_reconstructioninterface::reconstructionwithlvira3d Interface Reference	539
6.794 f_reconstructioninterface::reconstructionwithmof2d Interface Reference	539
6.795 f_reconstructioninterface::reconstructionwithmof3d Interface Reference	540
6.796 f_reconstructioninterface::reconstructionwithr2p2d Interface Reference	540
6.797 f_reconstructioninterface::reconstructionwithr2p2ddebug Interface Reference	540
6.798 f_reconstructioninterface::reconstructionwithr2p3d Interface Reference	540
6.799 f_reconstructioninterface::reconstructionwithr2p3ddebug Interface Reference	541
6.800 f_rectangularcuboid_class::rectangularcuboid_type Type Reference	541
6.801 IRL::RectangularCuboidBase< VertexType > Class Template Reference	541
6.802 IRL::RectangularCuboidCommon< Derived, VertexType > Class Template Reference	542
6.803 IRL::RectangularCuboidSpecialization< Derived, VertexType > Class Template Reference	542
6.803.1 Detailed Description	542
6.804 IRL::RecursiveSimplexCutting Struct Reference	543
6.805 IRL::ReferenceFrame Class Reference	543
6.805.1 Detailed Description	543
6.806 IRL::ReferredToPlaneOFExistence Class Reference	544
6.807 f_bytebuffer_class::resetbufferpointer Interface Reference	544
6.808 f_dividedpolygon_class::resetcentroid Interface Reference	545
6.809 IRL::RestrictedLocalizerLinkFromLocalizedSeparatorLink Class Reference	545

6.810 IRL::RestrictedMaskedLocalizedSeparatorLink Class Reference	546
6.811 f_tri_class::reverseptordering Interface Reference	547
6.812 f_dividedpolygon_class::reverseptordering Interface Reference	547
6.813 f_polygon_class::reverseptordering Interface Reference	547
6.814 IRL::Secant< OptimizingClass > Class Template Reference	548
6.814.1 Detailed Description	549
6.814.2 Member Function Documentation	549
6.814.2.1 solve() [1/2]	549
6.814.2.2 solve() [2/2]	549
6.814.2.3 solveAndTrackBounds()	550
6.814.3 Member Data Documentation	550
6.814.3.1 reason_for_exit_m	550
6.815 IRL::SegmentedDecomposedPolygon< VertexType > Class Template Reference	551
6.816 IRL::SegmentedDecomposedPolyhedron< VertexType > Class Template Reference	552
6.817 IRL::SegmentedDecomposedPolytope< VertexStorageType, VertexType, ProxyType > Class Template Reference	553
6.818 IRL::SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference	554
6.819 IRL::SegmentedHalfEdgePolygonCommon< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference	555
6.820 IRL::SegmentedHalfEdgePolygonSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference	556
6.821 IRL::SegmentedHalfEdgePolygonSpecificPt< Pt, FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference	556
6.822 IRL::SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference	556
6.823 IRL::SegmentedHalfEdgePolyhedronCommon< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference	557
6.824 IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference	558
6.825 IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference	559
6.826 IRL::SegmentedHalfEdgePolytope< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference	559
6.827 IRL::SelfExpandingCollection< ObjectType > Class Template Reference	561

6.827.1 Detailed Description	562
6.828 IRL::SeparatedMoments< MomentsType > Class Template Reference	562
6.829 IRL::SeparatedMoments< VolumeMoments > Class Template Reference	562
6.830 IRL::SeparatedMoments< VolumeMomentsAndDoubles< kArrayLength > > Class Template Reference	563
6.831 IRL::SeparatedMomentsCommon< Derived, MomentsType > Class Template Reference	564
6.831.1 Detailed Description	566
6.831.2 Constructor & Destructor Documentation	566
6.831.2.1 SeparatedMomentsCommon()	566
6.831.3 Member Function Documentation	566
6.831.3.1 fillWithComplementMoments()	567
6.832 f_sepvm_doubles3_class::sepvm_doubles3_type Type Reference	568
6.833 f_sepvm_class::sepvm_type Type Reference	568
6.834 f_serializer::serializeandpack Interface Reference	569
6.835 f_lviraneighborhood_rectangularcuboid_class::setcenterofstencil Interface Reference	569
6.836 f_r2pneighborhood_rectangularcuboid_class::setcenterofstencil Interface Reference	569
6.837 f_polyhedron24_doubles3_class::setdata Interface Reference	570
6.838 f_cappeddodecahedron_doubles3_class::setdata Interface Reference	570
6.839 f_volumefractionmatching::setdistancetomatchvolumefraction Interface Reference	570
6.840 f_localizedseparatorlink_class::setedgeconnectivity Interface Reference	570
6.841 f_localizerlink_class::setedgeconnectivity Interface Reference	571
6.842 f_localizedseparatorlink_class::setedgeconnectivitynull Interface Reference	571
6.843 f_localizerlink_class::setedgeconnectivitynull Interface Reference	571
6.844 f_planarlocalizer_class::setfromrectangularcuboid Interface Reference	571
6.845 f_localizerlink_class::setid Interface Reference	572
6.846 f_localizedseparatorlink_class::setid Interface Reference	572
6.847 f_elviraneighborhood_class::setmember Interface Reference	572
6.848 f_r2pneighborhood_rectangularcuboid_class::setmember Interface Reference	572
6.849 f_lviraneighborhood_rectangularcuboid_class::setmember Interface Reference	573
6.850 f_planarseparator_class::setnumberofplanes Interface Reference	573
6.851 f_planarlocalizer_class::setnumberofplanes Interface Reference	573

6.852 f_planarlocalizer_class::setplane Interface Reference	573
6.853 f_planarseparator_class::setplane Interface Reference	574
6.854 f_tri_class::setplaneofexistence Interface Reference	574
6.855 f_polygon_class::setplaneofexistence Interface Reference	574
6.856 f_dividedpolygon_class::setplaneofexistence Interface Reference	574
6.857 f_polyhedron24_doubles3_class::setpt Interface Reference	575
6.858 f_cappeddodecahedron_doubles3_class::setpt Interface Reference	575
6.859 f_polyhedron24_class::setpt Interface Reference	575
6.860 f_elviraneighborhood_class::setszie Interface Reference	575
6.861 f_lviraneighborhood_rectangularcuboid_class::setszie Interface Reference	576
6.862 f_r2pneighborhood_rectangularcuboid_class::setszie Interface Reference	576
6.863 f_bytebuffer_class::setszie Interface Reference	576
6.864 f_r2pneighborhood_rectangularcuboid_class::setsurfacearea Interface Reference	576
6.865 IRL::ShareVolumeAbovePlaneWithLinkedNeighbor Struct Reference	577
6.866 short_alloc::short_alloc< T, N, Align > Class Template Reference	577
6.867 IRL::SimplexCutting Struct Reference	578
6.868 IRL::SimplexWrapper< SimplexType, Enable > Struct Template Reference	578
6.869 IRL::SimplexWrapper< SimplexType, enable_if_t< is_tet< SimplexType >::value > > Struct Template Reference	578
6.869.1 Member Function Documentation	579
6.869.1.1 getSimplexVerticesAndIntersectionPoints()	579
6.869.1.2 simplexFromCutSimplexVertices()	579
6.870 IRL::SimplexWrapper< SimplexType, enable_if_t< is_tri< SimplexType >::value > > Struct Template Reference	580
6.871 IRL::SmallVector< T, BuffSizeInElements > Class Template Reference	580
6.872 IRL::StackVector< ObjectType, kMaxSize > Class Template Reference	582
6.873 IRL::StolenGraph< NodeType, StolenGraphType > Class Template Reference	583
6.873.1 Detailed Description	584
6.874 IRL::StoredCappedDodecahedron< VertexType > Class Template Reference	584
6.875 IRL::StoredConcaveBox< VertexType > Class Template Reference	585
6.876 IRL::StoredDodecahedron< VertexType > Class Template Reference	585

6.877 IRL::StoredHexahedron< VertexType > Class Template Reference	586
6.878 IRL::StoredPlaneOfExistence Class Reference	586
6.879 IRL::StoredPolyhedron24< VertexType > Class Template Reference	587
6.880 IRL::StoredRectangularCuboid< VertexType > Class Template Reference	588
6.881 IRL::StoredTet< VertexType > Class Template Reference	588
6.882 IRL::StoredTri< VertexType > Class Template Reference	589
6.883 IRL::StoredVertexAccess< Derived, VertexType, kNumberOfVertices > Class Template Reference	590
6.884 IRL::StrongWolfeConditionLineSearch Class Reference	591
6.884.1 Detailed Description	592
6.885 f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_type Type Reference	592
6.886 f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_type Type Reference	593
6.887 f_tagged_accumvm_vm_class::tagged_accumvm_vm_type Type Reference	594
6.888 IRL::TaggedAccumulatedListedVolumeMoments< VolumeMomentsType > Class Template Reference	594
6.889 IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType > Class Template Reference	595
6.890 IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType >::TaggedMoments Struct Reference	597
6.891 f_tet_class::tet_type Type Reference	597
6.892 IRL::TetSpecialization< Derived, VertexType > Class Template Reference	598
6.893 f_tri_class::tri_type Type Reference	599
6.894 IRL::TriSpecialization< Derived, VertexType > Class Template Reference	600
6.895 IRL::UnDirectedGraphNode< NodeType > Class Template Reference	602
6.895.1 Detailed Description	603
6.896 IRL::UnitQuaternion Class Reference	603
6.896.1 Detailed Description	604
6.897 f_serializer::unpackandstore Interface Reference	604
6.898 IRL::vector_sum< V1, V2 > Class Template Reference	604
6.899 IRL::Vertex< VertexLocationType > Class Template Reference	606
6.900 IRL::VertexList< VertexStorage, VertexType > Class Template Reference	607
6.901 IRL::VertexLocation< PtType > Class Template Reference	608
6.902 f_vman_class::vman_type Type Reference	608

6.903 IRL::Volume Class Reference	609
6.903.1 Detailed Description	610
6.904 IRL::Volume2D_Functor Class Reference	610
6.905 IRL::Volume3D_Functor Class Reference	611
6.906 IRL::VolumeMoments Class Reference	611
6.906.1 Detailed Description	613
6.906.2 Constructor & Destructor Documentation	613
6.906.2.1 VolumeMoments()	613
6.907 IRL::VolumeMoments2D_Functor Class Reference	614
6.908 IRL::VolumeMoments3D_Functor Class Reference	615
6.909 IRL::VolumeMomentsAndDoubles< kArrayLength > Class Template Reference	616
6.909.1 Constructor & Destructor Documentation	617
6.909.1.1 VolumeMomentsAndDoubles()	618
6.910 IRL::VolumeMomentsAndDoubles3D_Functor< kArrayLength > Class Template Reference	618
6.911 IRL::VolumeMomentsAndNormal Class Reference	619
6.911.1 Detailed Description	620
6.912 IRL::VolumeMomentsAndNormal2D_Functor Class Reference	621
6.913 f_listedvmm_vman_class::zeronormalcomponent Interface Reference	622
6.914 f_dividedpolygon_class::zeropolygong Interface Reference	622
6.915 f_polygon_class::zeropolygong Interface Reference	622

7 File Documentation	623
7.1 c_constants.h File Reference	623
7.1.1 Detailed Description	624
7.1.2 Function Documentation	624
7.1.2.1 c_Constants_setMinimumSurfaceAreaToTrack()	624
7.1.2.2 c_Constants_setMinimumVolumeToTrack()	624
7.1.2.3 c_Constants_setVolumeFractionBounds()	625
7.1.2.4 c_Constants_setVolumeFractionToleranceForIterativeDistanceFinding()	625
7.2 c_cut_polygon.h File Reference	625
7.2.1 Detailed Description	626
7.2.2 Function Documentation	626
7.2.2.1 c_getPlanePolygonFromReconstruction_RectangularCuboid_DividedPolygon()	626
7.2.2.2 c_getPlanePolygonFromReconstruction_RectangularCuboid_Polygon()	627
7.2.2.3 c_getReconstructionSurfaceArea_RectangularCuboid()	627
7.3 c_generic_cutting.h File Reference	627
7.3.1 Detailed Description	629
7.3.2 Function Documentation	630
7.3.2.1 c_getVolumeMoments_setMethod()	630
7.4 c_localizers.h File Reference	630
7.4.1 Detailed Description	631
7.5 c_serializer.h File Reference	631
7.5.1 Detailed Description	632
7.6 constants.h File Reference	632
7.6.1 Detailed Description	633
7.6.2 Variable Documentation	633
7.6.2.1 ARBITRARILY_LARGE_DISTANCE	633
7.6.2.2 MAX_PLANES_IN_OBJECT	634
7.7 f_bytebuffer_class.f90 File Reference	634
7.7.1 Detailed Description	635
7.8 f_cappeddodecahedron_class.f90 File Reference	635

7.8.1 Detailed Description	635
7.9 f_cappeddodecahedron_doubles3_class.f90 File Reference	636
7.9.1 Detailed Description	637
7.10 f_constants.f90 File Reference	637
7.10.1 Detailed Description	637
7.11 f_cutpolygon.f90 File Reference	637
7.11.1 Detailed Description	638
7.12 f_dividedpolygon_class.f90 File Reference	638
7.12.1 Detailed Description	640
7.13 f_dodecahedron_class.f90 File Reference	640
7.13.1 Detailed Description	640
7.14 f_geometriccuttinghelpers.f90 File Reference	640
7.14.1 Detailed Description	641
7.15 f_getvolumemoments.f90 File Reference	641
7.15.1 Detailed Description	642
7.16 f_localizedseparatorlink_class.f90 File Reference	642
7.16.1 Detailed Description	643
7.17 f_localizerlink_class.f90 File Reference	643
7.17.1 Detailed Description	644
7.18 f_objectallocationserver_localizedseparatorlink_class.f90 File Reference	644
7.18.1 Detailed Description	645
7.19 f_objectallocationserver_localizerlink_class.f90 File Reference	645
7.19.1 Detailed Description	646
7.20 f_objectallocationserver_planarlocalizer_class.f90 File Reference	646
7.20.1 Detailed Description	646
7.21 f_objectallocationserver_planarseparator_class.f90 File Reference	647
7.21.1 Detailed Description	647
7.22 f_planarlocalizer_class.f90 File Reference	647
7.22.1 Detailed Description	648
7.23 f_planarseparator_class.f90 File Reference	648

7.23.1 Detailed Description	649
7.24 f_polygon_class.f90 File Reference	649
7.24.1 Detailed Description	651
7.25 f_polyhedron24_class.f90 File Reference	651
7.25.1 Detailed Description	652
7.26 f_polyhedron24_doubles3_class.f90 File Reference	652
7.26.1 Detailed Description	653
7.27 f_r2pneighborhood_rectangularcuboid_class.f90 File Reference	653
7.27.1 Detailed Description	654
7.28 f_rectangularcuboid_class.f90 File Reference	654
7.28.1 Detailed Description	654
7.29 f_sepvm_class.f90 File Reference	655
7.29.1 Detailed Description	655
7.30 f_sepvm_doubles3_class.f90 File Reference	656
7.30.1 Detailed Description	657
7.31 f_serializer.f90 File Reference	657
7.31.1 Detailed Description	657
7.32 f_tagged_accumlistedvm_vman_class.f90 File Reference	657
7.32.1 Detailed Description	658
7.33 f_tagged_accumvm_sepvm_class.f90 File Reference	658
7.33.1 Detailed Description	660
7.34 f_tagged_accumvm_vm_class.f90 File Reference	660
7.34.1 Detailed Description	661
7.35 f_tet_class.f90 File Reference	661
7.35.1 Detailed Description	662
7.36 f_tri_class.f90 File Reference	662
7.36.1 Detailed Description	663
7.37 f_vman_class.f90 File Reference	663
7.37.1 Detailed Description	664
7.38 f_volumefractionmatching.f90 File Reference	664

7.38.1 Detailed Description	665
7.39 helper.h File Reference	665
7.39.1 Detailed Description	666
7.40 irl_fortran_interface.f90 File Reference	666
7.40.1 Detailed Description	667
7.41 lookup_tables.h File Reference	667
7.41.1 Detailed Description	669
7.41.2 Variable Documentation	669
7.41.2.1 cuboid_cut_vertices	669
7.41.2.2 cut_vertices [1/2]	670
7.41.2.3 cut_vertices [2/2]	670
7.41.2.4 number_of_negative_tets_after_cut	671
7.41.2.5 number_of_negative_tris_after_cut	671
7.41.2.6 number_of_new_vertices_after_cut [1/2]	671
7.41.2.7 number_of_new_vertices_after_cut [2/2]	672
7.41.2.8 number_of_plane_intersections_with_cuboid	672
7.41.2.9 number_of_tets_after_cut	672
7.41.2.10 number_of_tets_under_plane	673
7.41.2.11 number_of_tris_after_cut	673
7.41.2.12 tet_vertices	674
7.41.2.13 total_number_of_tets	674
7.41.2.14 verts_for_tets	674
7.41.2.15 verts_for_tris	675
7.42 lvira_optimization.h File Reference	675
7.42.1 Detailed Description	677
7.43 mymath.h File Reference	677
7.43.1 Detailed Description	678
7.44 plane_distance.h File Reference	678
7.44.1 Detailed Description	680
7.45 r2p_optimization.h File Reference	680
7.45.1 Detailed Description	682
7.46 rotations.h File Reference	682
7.46.1 Detailed Description	684
Index	685

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

f_bytebuffer_class	A fortran type class that allows the creation of IRL 's ByteBuffer class along with enabling some of its methods	55
f_cappeddodecahedron_class	A fortran type class that allows the creation of IRL 's CappedDodecahedron class along with enabling some of its methods	56
f_cappeddodecahedron_doubles3_class	A fortran type class that allows the creation of IRL 's CappedDodecahedron_doubles3 class along with enabling some of its methods	56
f_constants	This module contains mappings to the IRL C interface that deal with setting global constants that are used in the IRL library	57
f_cutpolygon	This module contains mappings to the IRL C interface that deal with intersecting planes to generate polygons and creating polygons that are representative of planar reconstructions in given cells	58
f_definedtypes	This module contains mappings to the IRL C interface that deal with intersecting planes to generate polygons and creating polygons that are representative of planar reconstructions in given cells	59
f_dividedpolygon_class	A fortran type class that allows the creation of IRL 's DividedPolygon class along with enabling some of its methods	59
f_dodecahedron_class	A fortran type class that allows the creation of IRL 's Dodecahedron class along with enabling some of its methods	61
f_elviraneighborhood_class	A fortran type class to provide the functionality of ELVIRANeighborhood	61
f_geometriccuttinghelpers	This module contains mappings to the IRL C interface that provides access to functions often used to geometric cutting operations. See the C interface file src/c_interface/c_geometric← cutting_helpers.h for more information	62
f_getvolumemoments	This module contains mappings to the IRL C interface that deal with intersecting polyhedron volumes and integrating these volumes to obtain volumetric moments	62

f_listedvm_vman_class	A fortran type class that allows the creation of IRL 's ListedVolumeMomentsM<VolumeMomentsAndNormal> class along with enabling some of its methods	64
f_localizedseparatorlink_class	A fortran type class that allows the creation of IRL 's LocalizedSeparatorLink class along with enabling some of its methods	65
f_localizerlink_class	A fortran type class that allows the creation of IRL 's LocalizerLink class along with enabling some of its methods	65
f_lviraneighborhood_rectangularcuboid_class	A fortran type class to provide the functionality of LVIRANeighborhood	66
f_objectallocationserver_localizedseparatorlink_class	A fortran type class that allows the creation of IRL 's ObjectAllocationServer<LocalizedSeparatorLink> class along with enabling some of its methods	67
f_objectallocationserver_localizerlink_class	A fortran type class that allows the creation of IRL 's ObjectAllocationServer<LocalizerLink> class along with enabling some of its methods	68
f_objectallocationserver_planarlocalizer_class	A fortran type class that allows the creation of IRL 's ObjectAllocationServer<PlanarLocalizer> class along with enabling some of its methods	68
f_objectallocationserver_planarseparator_class	A fortran type class that allows the creation of IRL 's ObjectAllocationServer<PlanarSeparator> class along with enabling some of its methods	69
f_planarlocalizer_class	A fortran type class that allows the creation of IRL 's PlanarLocalizer class along with enabling some of its methods	69
f_planarseparator_class	A fortran type class that allows the creation of IRL 's PlanarSeparator class along with enabling some of its methods	70
f_polygon_class	A fortran type class that allows the creation of IRL 's Polygon class along with enabling some of its methods	71
f_polyhedron24_class	A fortran type class that allows the creation of IRL 's Polyhedron24 class along with enabling some of its methods	72
f_polyhedron24_doubles3_class	A fortran type class that allows the creation of IRL 's Polyhedron24_doubles3 class along with enabling some of its methods	73
f_r2pneighborhood_rectangularcuboid_class	A fortran type class to provide the functionality of R2PNeighborhood_RectangularCuboid	74
f_reconstructioninterface	This module contains interface reconstruction methods that can be used to obtain PlanarSeparators. The requirements to use each type of reconstruction are different. Please consult the documentation and examples before using a specific reconstruction type	75
f_rectangularcuboid_class	A fortran type class that allows the creation of IRL 's RectangularCuboid class along with enabling some of its methods	77
f_sepvm_class	A fortran type class that allows the creation of IRL 's SeparatedMoments<VolumeMoments> class along with enabling some of its methods	77
f_sepvm_doubles3_class	A fortran type class that allows the creation of IRL 's SeparatedMoments<VolumeMoments> class along with enabling some of its methods	78
f_serializer	This module contains mappings to the IRL C interface that deal with serializing IRL class objects into an array of bytes and packing them into a byte buffer	79

f_tagged_accumlistedvm_vman_class	A fortran type class that allows the creation of IRL 's TaggedAccumulatedListedVolumeMomentsM<VolumeMomentsAndNormal> class along with enabling some of its methods	80
f_tagged_accumvm_sepvm_class	A fortran type class that allows the creation of IRL 's AccumulatedVolumeMomentsM<SeparatedVolumeMoments>> class along with enabling some of its methods	81
f_tagged_accumvm_vm_class	A fortran type class that allows the creation of IRL 's AccumulatedVolumeMomentsM<VolumeMoments> class along with enabling some of its methods	82
f_tet_class	A fortran type class that allows the creation of IRL 's Tet class along with enabling some of its methods	83
f_tri_class	A fortran type class that allows the creation of IRL 's Tri class along with enabling some of its methods	84
f_vman_class	A fortran type class that allows the creation of IRL 's AccumulatedListedVolumeMomentsM<VolumeMomentsAndNormal> class along with enabling some of its methods	85
f_volumefractionmatching	This module contains mappings to the IRL C interface that deals with setting the distance to each plane in a reconstruction to recreate the volume fraction on the provided polyhedron	85
IRL	Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions	86
irl_fortran_interface	This is just a master wrapper for the entire IRL fortran interface. For information about each module, view the documentation for the module itself	117

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

IRL::AccumulatedVolumeMoments< VolumeMomentsType >	121
IRL::AccumulatedVolumeMoments< ListedVolumeMoments< VolumeMomentsType > >	121
IRL::AccumulatedListedVolumeMoments< VolumeMomentsType >	119
IRL::AccumulateIntoCollection	122
IRL::AccumulateIntoScalar	123
f_lviraneighborhood_rectangularcuboid_class::addmember	125
f_r2pnighborhood_rectangularcuboid_class::addmember	126
f_planarseparator_class::addplane	126
f_planarlocalizer_class::addplane	126
f_cappeddodecahedron_doubles3_class::adjustcaptomatchvolume	126
f_cappeddodecahedron_class::adjustcaptomatchvolume	127
f_polyhedron24_class::adjustcaptomatchvolume	127
f_polyhedron24_doubles3_class::adjustcaptomatchvolume	127
IRL::AdvectedPlaneReconstruction	127
IRL::AdvectedPlaneReconstructionDebug	128
f_listedvm_vman_class::append	129
f_tagged_accumlistedvm_vman_class::append	129
short_alloc::arena< N, alignment >	129
IRL::BFGS< OptimizingClass, kParameters >	130
IRL::Bisection< OptimizingClass >	131
IRL::BlockObjectAllocation< ObjectType >	133
IRL::BlockObjectAllocation< IRL::PlanarLocalizer >	133
IRL::BlockObjectAllocation< IRL::PlanarSeparator >	133
IRL::BlockObjectAllocation< IRL::ReconstructionLink >	133
IRL::BrentsMethod< OptimizingClass >	134
IRL::ByteBuffer	136
f_bytебuffer_class::bytebuffer_type	137
f_bytебuffer_class::c_bytебuffer	137
c(ByteBuffer	138
c(CappedDodecahedron	138
f_cappeddodecahedron_class::c_cappeddodecahedron	139
c(CappedDodecahedron_doubles3	139
f_cappeddodecahedron_doubles3_class::c_cappeddodecahedron_doubles3	139
c(DividedPolygon	140
f_dividedpolygon_class::c_dividedpolygon	140

f_dodecahedron_class::c_dodecahedron	140
c_Dodecahedron	140
c_ELVIRANeighborhood	141
f_elviraneighborhood_class::c_elviraneighborhood	142
c_ListedVM_VMAN	142
f_listedvm_vman_class::c_listedvm_vman	143
c_LocalizedSeparatorLink	143
f_localizedseparatorlink_class::c_localizedseparatorlink	143
c_LocalizerLink	144
f_localizerlink_class::c_localizerlink	144
c_LVIRANeighborhood_RectangularCuboid	145
f_lviraneighborhood_rectangularcuboid_class::c_lviraneighborhood_rectangularcuboid	145
c_ObjectAllocationServer_LocalizedSeparatorLink	146
f_objectallocationserver_localizedseparatorlink_class::c_objectallocationserver_localizedseparatorlink	146
f_objectallocationserver_localizerlink_class::c_objectallocationserver_localizerlink	146
c_ObjectAllocationServer_LocalizerLink	147
f_objectallocationserver_planarlocalizer_class::c_objectallocationserver_planarlocalizer	147
c_ObjectAllocationServer_PlanarLocalizer	148
c_ObjectAllocationServer_PlanarSeparator	149
f_objectallocationserver_planarseparator_class::c_objectallocationserver_planarseparator	150
f_planarlocalizer_class::c_planarlocalizer	150
c_PlanarLocalizer	150
f_planarseparator_class::c_planarseparator	151
c_PlanarSeparator	151
f_polygon_class::c_polygon	152
c_Polygon	152
f_polyhedron24_class::c_polyhedron24	152
c_Polyhedron24	153
f_polyhedron24_doubles3_class::c_polyhedron24_doubles3	153
c_Polyhedron24_doubles3	153
c_R2PNeighborhood_RectangularCuboid	154
f_r2pneighborhood_rectangularcuboid_class::c_r2pneighborhood_rectangularcuboid	154
c_RectangularCuboid	155
f_rectangularcuboid_class::c_rectangularcuboid	155
c_SepVM	155
f_sepvm_class::c_sepvm	156
c_SepVM_doubles3	156
f_sepvm_doubles3_class::c_sepvm_doubles3	157
c_Tagged_AccumListedVM_VMAN	157
f_tagged_accumlistedvm_vman_class::c_tagged_accumlistedvm_vman	157
f_tagged_accumvm_sepvm_class::c_tagged_accumvm_sepvm	158
c_Tagged_AccumVM_SepVM	158
f_tagged_accumvm_vm_class::c_tagged_accumvm_vm	158
c_Tagged_AccumVM_VM	159
f_tet_class::c_tet	159
c_Tet	159
f_tri_class::c_tri	160
c_Tri	160
f_vman_class::c_vman	160
c_VMAN	161
f_tri_class::calculateandsetplaneofexistence	161
f_polygon_class::calculateandsetplaneofexistence	162
f_dividedpolygon_class::calculateandsetplaneofexistence	162
f_tri_class::calculatecentroid	162
f_polygon_class::calculatecentroid	162
f_polygon_class::calculatenarestptonsurface	163
f_tri_class::calculatenormal	163
f_polygon_class::calculatenormal	163

f_dividedpolygon_class::calculatenormal	163
f_tri_class::calculatesign	164
f_polygon_class::calculatesign	164
f_dividedpolygon_class::calculatesign	164
f_dividedpolygon_class::calculatesurfacearea	164
f_tri_class::calculatevolume	165
f_polygon_class::calculatevolume	165
f_rectangularcuboid_class::calculatevolume	165
f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_type	165
f_cappeddodecahedron_class::cappeddodecahedron_type	166
IRL::CellCollection< CellGroupType >	168
IRL::CellCollection< IRL::CellGroupedMoments >	168
IRL::CellGroupedMoments< CellType, ContainedMomentsType >	169
IRL::CellGroupedMoments< CellType, IRL::SeparatedMoments< IRL::VolumeMoments > >	169
IRL::Centroid2D_Functor	171
IRL::Centroid3D_Functor	172
IRL::ChainedBlockStorage< ContainedType, kBlockSize >	173
IRL::ChainedBlockStorage< Face< HalfEdge< Vertex< VertexLocation< PtType > > >, half_edge← _polytope::default_sizes::complete_block_faces >	173
IRL::ChainedBlockStorage< FaceType, kMaxFaces >	173
IRL::ChainedBlockStorage< HalfEdge< Vertex< VertexLocation< PtType > > >, half_edge_polytope← ::default_sizes::complete_block_half_edges >	173
IRL::ChainedBlockStorage< HalfEdgeType, kMaxHalfEdges >	173
IRL::ChainedBlockStorage< Vertex< VertexLocation< PtType > >, half_edge_polytope::default_sizes← ::complete_block_vertices >	173
IRL::ChainedBlockStorage< VertexLocation< PtType >, half_edge_polytope::default_sizes::complete← _block_vertices >	173
IRL::ChainedBlockStorage< VertexLocationType, kMaxVertices >	173
IRL::ChainedBlockStorage< VertexType, kMaxVertices >	173
f_listedvm_vman_class::clear	174
f_tagged_accumlistedvm_vman_class::clear	174
IRL::Collection< ObjectType >	174
IRL::Lister< ObjectType >	433
IRL::SelfExpandingCollection< ObjectType >	561
IRL::Accumulator< ObjectType >	124
IRL::Collection< CellGroupType >	174
IRL::Collection< IRL::CellGroupedMoments >	174
IRL::Collection< IRL::VolumeMomentsAndNormal >	174
IRL::Lister< IRL::VolumeMomentsAndNormal >	433
IRL::Collection< ListedVolumeMoments< VolumeMomentsType > >	174
IRL::SelfExpandingCollection< ListedVolumeMoments< VolumeMomentsType > >	561
IRL::Accumulator< ListedVolumeMoments< VolumeMomentsType > >	124
IRL::Collection< VolumeMomentsType >	174
IRL::Lister< VolumeMomentsType >	433
IRL::SelfExpandingCollection< VolumeMomentsType >	561
IRL::Accumulator< VolumeMomentsType >	124
f_tet_class::construct	178
f_tri_class::construct	178
f_cappeddodecahedron_class::construct	178
f_dodecahedron_class::construct	179
f_cappeddodecahedron_doubles3_class::construct	179
f_polygon_class::construct	179
f_polyhedron24_class::construct	179
f_polyhedron24_doubles3_class::construct	180
f_dividedpolygon_class::construct	180
f_rectangularcuboid_class::construct	180
f_sepvm_class::construct	180

f_rectangularcuboid_class::construct_2pt	181
f_dividedpolygon_class::constructfrompolygon	181
IRL::ContinueDividingVolumeByPlanarReconstruction< HowToCutSimplexByNextPlane, HowToBranch< EarlyIfNoIntersection, HowToGenerateAndHandleSimplicesFromVolumeUnderPlane, HowToGenerateAndHandleSimplicesFromVolumeAbovePlane >	181
IRL::ContinueReducingVolumeToBeInternalToReconstruction	182
f_planarseparator_class::copy	182
IRL::CutSimplexByNextPlaneAccessedByIndex	182
IRL::CutTetVertices< VertexType >	182
IRL::CutTriangleVerticesAndPlane< VertexType >	183
f_bytebuffer_class::dataptr	184
IRL::DecomposedPolytopeVertexStorage< VertexType >	187
IRL::DecomposedPolygonVertexStorage< VertexType >	184
IRL::DecomposedPolyhedronVertexStorage< VertexType >	186
IRL::DelayedExpandableVertexAccess< Derived, VertexType, kMaskedVertices >	188
f_dividedpolygon_class::dividedpolygon_type	189
f_dodecahedron_class::dodecahedron_type	192
IRL::DoesNotHaveACollection< ReturnType >	193
IRL::DoesNotHaveALocalizer< ReconstructionType >	193
IRL::DoesNotHaveANestedType< ReturnType >	194
IRL::DoesNotHaveAReconstructionLink< ReconstructionType >	194
IRL::DoesNotHaveASeparator< ReconstructionType >	194
IRL::doNoEarlyBranching	195
IRL::earlyBranchBothFullyBelowAndFullyAboveSimplicesLink	195
IRL::earlyBranchBothFullyBelowAndFullyAboveSimplicesNoLink	195
IRL::ELVIRA_2D	196
IRL::ELVIRADebug< ELVIRA_2D >	199
IRL::ELVIRA_3D	197
IRL::ELVIRADebug< ELVIRA_3D >	200
IRL::ELVIRADebug< BaseELVIRA >	199
IRL::ELVIRANeighborhood	201
f_elviraneighborhood_class::elviraneighborhood_type	203
f_r2pneighborhood_rectangularcuboid_class::emptyneighborhood	203
f_lviraneighborhood_rectangularcuboid_class::emptyneighborhood	204
f_listedvmm_vman_class::erase	204
IRL::ExpandablePtList< VertexType >	206
IRL::Expr< E >	208
IRL::Expr< Pt >	208
IRL::Pt	509
IRL::Expr< vector_sum< V1, V2 > >	208
IRL::vector_sum< V1, V2 >	604
f_bytebuffer_class::F_ByteBuffer_dataPtr	208
f_bytebuffer_class::F_ByteBuffer_delete	208
f_bytebuffer_class::F_ByteBuffer_getSize	208
f_bytebuffer_class::F_ByteBuffer_new	209
f_bytebuffer_class::F_ByteBuffer_resetBufferPointer	209
f_bytebuffer_class::F_ByteBuffer_setSize	209
f_cappeddodecahedron_class::F_CappedDodecahedron_adjustCapToMatchVolume	209
f_cappeddodecahedron_class::F_CappedDodecahedron_construct	210
f_cappeddodecahedron_class::F_CappedDodecahedron_delete	210
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_adjustCapToMatchVolume	210
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_construct	210
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_delete	211
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getBoundingPts	211
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getData	211
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getPt	211
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_new	212

f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_setData	212
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_setPt	212
f_cappeddodecahedron_class::F_CappedDodecahedron_getBoundingPts	212
f_cappeddodecahedron_class::F_CappedDodecahedron_getPt	213
f_cappeddodecahedron_class::F_CappedDodecahedron_new	213
f_constants::F_Constants_setMinimumSurfaceAreaToTrack	213
f_constants::F_Constants_setMinimumVolumeToTrack	213
f_constants::F_Constants_setVolumeFractionBounds	214
f_constants::F_Constants_setVolumeFractionToleranceForDistanceFinding	214
f_dividedpolygon_class::F_DividedPolygon_calculateAndSetPlaneOfExistence	214
f_dividedpolygon_class::F_DividedPolygon_calculateNormal	214
f_dividedpolygon_class::F_DividedPolygon_calculateSign	215
f_dividedpolygon_class::F_DividedPolygon_calculateSurfaceArea	215
f_dividedpolygon_class::F_DividedPolygon_construct	215
f_dividedpolygon_class::F_DividedPolygon_constructFromPolygon	215
f_dividedpolygon_class::F_DividedPolygon_delete	216
f_dividedpolygon_class::F_DividedPolygon_getBoundingPts	216
f_dividedpolygon_class::F_DividedPolygon_getLocalizer	216
f_dividedpolygon_class::F_DividedPolygon_getNumberOfPts	216
f_dividedpolygon_class::F_DividedPolygon_getNumberOfSimplicesInDecomposition	217
f_dividedpolygon_class::F_DividedPolygon_getPlaneOfExistence	217
f_dividedpolygon_class::F_DividedPolygon_getPt	217
f_dividedpolygon_class::F_DividedPolygon_getSimplexFromDecomposition	217
f_dividedpolygon_class::F_DividedPolygon_new	218
f_dividedpolygon_class::F_DividedPolygon_printToScreen	218
f_dividedpolygon_class::F_DividedPolygon_resetCentroid	218
f_dividedpolygon_class::F_DividedPolygon_reversePtOrdering	218
f_dividedpolygon_class::F_DividedPolygon_setPlaneOfExistence	219
f_dividedpolygon_class::F_DividedPolygon_zeroPolygon	219
f_dodecahedron_class::F_Dodecahedron_construct	219
f_dodecahedron_class::F_Dodecahedron_delete	219
f_dodecahedron_class::F_Dodecahedron_getBoundingPts	220
f_dodecahedron_class::F_Dodecahedron_new	220
f_elviraneighborhood_class::F_ELVIRANeighborhood_delete	220
f_elviraneighborhood_class::F_ELVIRANeighborhood_new	220
f_elviraneighborhood_class::F_ELVIRANeighborhood_setMember	221
f_elviraneighborhood_class::F_ELVIRANeighborhood_setSize	221
f_cutpolygon::F_getPlanePolygonFromReconstruction_RC_DivPoly	221
f_cutpolygon::F_getPlanePolygonFromReconstruction_RC_Poly	221
f_cutpolygon::F_getReconstructionSurfaceArea_RC	222
f_getvolumemoments::F_GNVM_CD_By_LSL_For_SVM	222
f_getvolumemoments::F_GNVM_CD_By_LSL_For_TagAccumVM_SVM	222
f_getvolumemoments::F_GNVM_CDWD3_By_LSL_For_SVMAD3	222
f_getvolumemoments::F_GNVM_D_By_LSL_For_SVM	223
f_getvolumemoments::F_GNVM_D_By_LSL_For_TagAccumVM_SVM	223
f_getvolumemoments::F_GNVM_D_By_PS_For_SVM	223
f_getvolumemoments::F_GNVM_P24_By_LSL_For_SVM	223
f_getvolumemoments::F_GNVM_P24WD3_By_LSL_For_SVMAD3	224
f_getvolumemoments::F_GNVM_Poly_By_PL_For_V	224
f_getvolumemoments::F_GNVM_RC_By_PS_For_SVM	224
f_getvolumemoments::F_GNVM_RC_By_PS_For_V	224
f_getvolumemoments::F_GNVM_Tet_By_LSL_For_SVM	225
f_getvolumemoments::F_GNVM_Tri_By_LL_For_TagAVM_VM	225
f_getvolumemoments::F_GNVM_Tri_By_PL_For_V	225
f_getvolumemoments::F_GVM_CD_By_LSL_For_SVM	225
f_getvolumemoments::F_GVM_D_By_LSL_For_SVM	226
f_getvolumemoments::F_GVM_P24_By_LSL_For_SVM	226
f_getvolumemoments::F_GVM_setMethod	226

f_getvolumemoments::F_GVM_Tri_By_LL_For_TagALVM_VMAN	226
f_geometriccuttinghelpers::F_isPtInternal_PL	227
f_geometriccuttinghelpers::F_isPtInternal_PS	227
f_listedvm_vman_class::F_ListedVM_VMAN_append	227
f_listedvm_vman_class::F_ListedVM_VMAN_clear	227
f_listedvm_vman_class::F_ListedVM_VMAN_delete	228
f_listedvm_vman_class::F_ListedVM_VMAN_erase	228
f_listedvm_vman_class::F_ListedVM_VMAN_getMoments	228
f_listedvm_vman_class::F_ListedVM_VMAN_getSize	228
f_listedvm_vman_class::F_ListedVM_VMAN_new	229
f_listedvm_vman_class::F_ListedVM_VMAN_zeroNormalComponent	229
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_delete	229
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_getId	229
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_new	230
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_newFromObjectAllocationServer	230
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivity	230
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivityNull	230
f_localizedseparatorlink_class::F_LocalizedSeparatorLink setId	231
f_localizerlink_class::F_LocalizerLink_delete	231
f_localizerlink_class::F_LocalizerLink_getId	231
f_localizerlink_class::F_LocalizerLink_new	231
f_localizerlink_class::F_LocalizerLink_newFromObjectAllocationServer	232
f_localizerlink_class::F_LocalizerLink_setEdgeConnectivity	232
f_localizerlink_class::F_LocalizerLink_setEdgeConnectivityNull	232
f_localizerlink_class::F_LocalizerLink setId	232
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_addMember	233
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_delete	233
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_empty← Neighborhood	233
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_new	233
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_setCenter← OfStencil	234
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_setMember	234
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_setSize	234
f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparator← Link_delete	234
f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparator← Link_new	235
f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_delete	235
f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_new	235
f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_delete	235
f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_new	236
f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_PlanarSeparator_delete	236
f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_PlanarSeparator_new	236
f_planarlocalizer_class::F_PlanarLocalizer_addPlane	236
f_planarlocalizer_class::F_PlanarLocalizer_delete	237
f_planarlocalizer_class::F_PlanarLocalizer_new	237
f_planarlocalizer_class::F_PlanarLocalizer_newFromObjectAllocationServer	237
f_planarlocalizer_class::F_PlanarLocalizer_printToScreen	237
f_planarlocalizer_class::F_PlanarLocalizer_setFromRectangularCuboid	238
f_planarlocalizer_class::F_PlanarLocalizer_setNumberOfPlanes	238
f_planarlocalizer_class::F_PlanarLocalizer_setPlane	238
f_planarseparator_class::F_PlanarSeparator_addPlane	238
f_planarseparator_class::F_PlanarSeparator_copy	239
f_planarseparator_class::F_PlanarSeparator_delete	239
f_planarseparator_class::F_PlanarSeparator_getNumberOfPlanes	239
f_planarseparator_class::F_PlanarSeparator_getPlane	239
f_planarseparator_class::F_PlanarSeparator_isFlipped	240

f_planarseparator_class::F_PlanarSeparator_new	240
f_planarseparator_class::F_PlanarSeparator_newFromObjectAllocationServer	240
f_planarseparator_class::F_PlanarSeparator_printToScreen	240
f_planarseparator_class::F_PlanarSeparator_setNumberOfPlanes	241
f_planarseparator_class::F_PlanarSeparator_setPlane	241
f_polygon_class::F_Polygon_calculateAndSetPlaneOfExistence	241
f_polygon_class::F_Polygon_calculateCentroid	241
f_polygon_class::F_Polygon_calculateNearestPtOnSurface	242
f_polygon_class::F_Polygon_calculateNormal	242
f_polygon_class::F_Polygon_calculateSign	242
f_polygon_class::F_Polygon_calculateVolume	242
f_polygon_class::F_Polygon_construct	243
f_polygon_class::F_Polygon_delete	243
f_polygon_class::F_Polygon_getBoundingPts	243
f_polygon_class::F_Polygon_getLocalizer	243
f_polygon_class::F_Polygon_getNumberOfPts	244
f_polygon_class::F_Polygon_getNumberOfSimplicesInDecomposition	244
f_polygon_class::F_Polygon_getPlaneOfExistence	244
f_polygon_class::F_Polygon_getPt	244
f_polygon_class::F_Polygon_getSimplexFromDecomposition	245
f_polygon_class::F_Polygon_new	245
f_polygon_class::F_Polygon_printToScreen	245
f_polygon_class::F_Polygon_reversePtOrdering	245
f_polygon_class::F_Polygon_setPlaneOfExistence	246
f_polygon_class::F_Polygon_zeroPolygon	246
f_polyhedron24_class::F_Polyhedron24_adjustCapToMatchVolume	246
f_polyhedron24_class::F_Polyhedron24_construct	246
f_polyhedron24_class::F_Polyhedron24_delete	247
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_adjustCapToMatchVolume	247
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_construct	247
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_delete	247
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getBoundingPts	248
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getData	248
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getPt	248
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_new	248
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setData	249
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setPt	249
f_polyhedron24_class::F_Polyhedron24_getBoundingPts	249
f_polyhedron24_class::F_Polyhedron24_getPt	249
f_polyhedron24_class::F_Polyhedron24_new	250
f_polyhedron24_class::F_Polyhedron24_setPt	250
f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_addMember	250
f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_delete	250
f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_emptyNeighborhood	251
f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_new	251
f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setCenterOf← Stencil	251
f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setMember	251
f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setSize	252
f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setSurfaceArea	252
f_reconstructioninterface::F_reconstructionWithAdvectionNormals_ListedVM_VMAN_RC	252
f_reconstructioninterface::F_reconstructionWithAdvectionNormalsDebug_ListedVM_VMAN_RC	252
f_reconstructioninterface::F_reconstructionWithELVIRA2D	253
f_reconstructioninterface::F_reconstructionWithELVIRA3D	253
f_reconstructioninterface::F_reconstructionWithLVIRA2D_RC	253
f_reconstructioninterface::F_reconstructionWithLVIRA3D_RC	253
f_reconstructioninterface::F_reconstructionWithMOF2D_RectangularCuboid	254
f_reconstructioninterface::F_reconstructionWithMOF2D_Tri	254

f_reconstructioninterface::F_reconstructionWithMOF2DGiveWeights_RectangularCuboid	254
f_reconstructioninterface::F_reconstructionWithMOF2DGiveWeights_Tri	254
f_reconstructioninterface::F_reconstructionWithMOF3D_RectangularCuboid	255
f_reconstructioninterface::F_reconstructionWithMOF3D_Tet	255
f_reconstructioninterface::F_reconstructionWithMOF3DGiveWeights_RectangularCuboid	255
f_reconstructioninterface::F_reconstructionWithMOF3DGiveWeights_Tet	255
f_reconstructioninterface::F_reconstructionWithR2P2D_RC	256
f_reconstructioninterface::F_reconstructionWithR2P2DDebug_RC	256
f_reconstructioninterface::F_reconstructionWithR2P3D_RC	256
f_reconstructioninterface::F_reconstructionWithR2P3DDebug_RC	256
f_rectangularcuboid_class::F_RectangularCuboid_calculateVolume	257
f_rectangularcuboid_class::F_RectangularCuboid_construct	257
f_rectangularcuboid_class::F_RectangularCuboid_construct_2pt	257
f_rectangularcuboid_class::F_RectangularCuboid_delete	257
f_rectangularcuboid_class::F_RectangularCuboid_getBoundingPts	258
f_rectangularcuboid_class::F_RectangularCuboid_new	258
f_sepvm_class::F_SepVM_construct	258
f_sepvm_class::F_SepVM_delete	258
f_sepvm_doubles3_class::F_SepVM_doubles3_delete	259
f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroid	259
f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroidPtr	259
f_sepvm_doubles3_class::F_SepVM_doubles3_getData	259
f_sepvm_doubles3_class::F_SepVM_doubles3_getVolume	260
f_sepvm_doubles3_class::F_SepVM_doubles3_getVolumePtr	260
f_sepvm_doubles3_class::F_SepVM_doubles3_multiplyByVolume	260
f_sepvm_doubles3_class::F_SepVM_doubles3_new	260
f_sepvm_doubles3_class::F_SepVM_doubles3_normalizeByVolume	261
f_sepvm_class::F_SepVM_getCentroid	261
f_sepvm_class::F_SepVM_getCentroidPtr	261
f_sepvm_class::F_SepVM_getVolume	261
f_sepvm_class::F_SepVM_getVolumePtr	262
f_sepvm_class::F_SepVM_multiplyByVolume	262
f_sepvm_class::F_SepVM_new	262
f_sepvm_class::F_SepVM_normalizeByVolume	262
f_serializer::F_Serializer_serializeAndPack_PlanarSeparator_ByteBuffer	263
f_serializer::F_Serializer_unpackAndStore_PlanarSeparator_ByteBuffer	263
f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS	263
f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS_DefTol	263
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_append	264
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_clear	264
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_delete	264
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getListAtIndex	264
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getSize	265
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getTagForIndex	265
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_new	265
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_delete	265
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtIndex	266
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtTag	266
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidPtrAtIndex	266
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getSize	266
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getTagForIndex	267
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtIndex	267
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtTag	267
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumePtrAtIndex	267
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_multiplyByVolume	268
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_new	268
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_normalizeByVolume	268
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_delete	268

f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidAtIndex	269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidPtrAtIndex	269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getSize	269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getTagForIndex	269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumeAtIndex	270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumePtrAtIndex	270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_multiplyByVolume	270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_new	270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_normalizeByVolume	271
f_tet_class::F_Tet_construct	271
f_tet_class::F_Tet_delete	271
f_tet_class::F_Tet_getBoundingPts	271
f_tet_class::F_Tet_new	272
f_tri_class::F_Tri_calculateAndSetPlaneOfExistence	272
f_tri_class::F_Tri_calculateCentroid	272
f_tri_class::F_Tri_calculateNormal	272
f_tri_class::F_Tri_calculateSign	273
f_tri_class::F_Tri_calculateVolume	273
f_tri_class::F_Tri_construct	273
f_tri_class::F_Tri_delete	273
f_tri_class::F_Tri_getBoundingPts	274
f_tri_class::F_Tri_getLocalizer	274
f_tri_class::F_Tri_getPlaneOfExistence	274
f_tri_class::F_Tri_getVertices	274
f_tri_class::F_Tri_new	275
f_tri_class::F_Tri_reversePtOrdering	275
f_tri_class::F_Tri_setPlaneOfExistence	275
f_vman_class::F_VMAN_delete	275
f_vman_class::F_VMAN_getCentroid	276
f_vman_class::F_VMAN_getNormal	276
f_vman_class::F_VMAN_getVolume	276
f_vman_class::F_VMAN_multiplyByVolume	276
f_vman_class::F_VMAN_new	277
f_vman_class::F_VMAN_normalizeByVolume	277
IRL::Face< HalfEdgeType >	277
IRL::Face< IRL::HalfEdge >	277
false_type	
IRL::has_localizer< C >	322
IRL::has_localizer< const C >	323
IRL::has_null_reconstruction< C >	334
IRL::has_null_reconstruction< const C >	335
IRL::has_separator< C >	337
IRL::has_separator< const C >	338
IRL::is_moments_collection< C >	353
IRL::is_moments_collection< const C >	357
IRL::is_nested_moments< C >	360
IRL::is_nested_moments< const C >	363
IRL::is_planar_separator< C >	364
IRL::is_planar_separator< const C >	365
IRL::is_polygon< C >	367
IRL::is_polygon< const C >	368
IRL::is_polyhedron< C >	375
IRL::is_polyhedron< const C >	376
IRL::is_reconstruction_link< C >	387
IRL::is_reconstruction_link< const C >	388
IRL::is_separated_moments< C >	396
IRL::is_separated_moments< const C >	398

IRL::is_tet< C >	400
IRL::is_tet< const C >	401
IRL::is_tri< C >	404
IRL::is_tri< const C >	405
IRL::isHalfEdgeCutting< C >	408
IRL::isHalfEdgeCutting< const C >	409
IRL::isRecursiveSimplexCutting< C >	412
IRL::isRecursiveSimplexCutting< const C >	413
IRL::isSimplexCutting< C >	415
IRL::isSimplexCutting< const C >	416
f_cappeddodecahedron_class::getboundingpts	283
f_tet_class::getboundingpts	283
f_tri_class::getboundingpts	284
f_dodecahedron_class::getboundingpts	284
f_cappeddodecahedron_doubles3_class::getboundingpts	284
f_polygon_class::getboundingpts	284
f_polyhedron24_class::getboundingpts	285
f_polyhedron24_doubles3_class::getboundingpts	285
f_rectangularcuboid_class::getboundingpts	285
f_dividedpolygon_class::getboundingpts	285
f_vman_class::getcentroid	286
f_sepvm_class::getcentroid	286
f_sepvm_doubles3_class::getcentroid	286
f_tagged_accumvm_sepvm_class::getcentroidatindex	286
f_tagged_accumvm_vm_class::getcentroidatindex	287
f_tagged_accumvm_sepvm_class::getcentroidattag	287
f_sepvm_class::getcentroidptr	287
f_sepvm_doubles3_class::getcentroidptr	287
f_tagged_accumvm_vm_class::getcentroidptratindex	288
f_tagged_accumvm_sepvm_class::getcentroidptratindex	288
f_objectallocationserver_localizedseparatorlink_class::getcobject	288
f_tagged_accumvm_sepvm_class::getcobject	288
f_objectallocationserver_localizerlink_class::getcobject	289
f_objectallocationserver_planarlocalizer_class::getcobject	289
f_tagged_accumvm_vm_class::getcobject	289
f_objectallocationserver_planarseparator_class::getcobject	289
f_planarlocalizer_class::getcobject	290
f_tet_class::getcobject	290
f_tri_class::getcobject	290
f_planarseparator_class::getcobject	290
f_dodecahedron_class::getcobject	291
f_vman_class::getcobject	291
f_cappeddodecahedron_doubles3_class::getcobject	291
f_elviraneighborhood_class::getcobject	291
f_polygon_class::getcobject	292
f_polyhedron24_class::getcobject	292
f_polyhedron24_doubles3_class::getcobject	292
f_listedvm_vman_class::getcobject	292
f_bytbuffer_class::getcobject	293
f_r2neighborhood_rectangularcuboid_class::getcobject	293
f_localizedseparatorlink_class::getcobject	293
f_dividedpolygon_class::getcobject	293
f_rectangularcuboid_class::getcobject	294
f_localizerlink_class::getcobject	294
f_sepvm_class::getcobject	294
f_sepvm_doubles3_class::getcobject	294
f_lviraneighborhood_rectangularcuboid_class::getcobject	295
f_tagged_accumlistedvm_vman_class::getcobject	295

f_cappeddodecahedron_class::getcobject	295
f_cappeddodecahedron_doubles3_class::getdata	295
f_polyhedron24_doubles3_class::getdata	296
f_sepvm_doubles3_class::getdata	296
f_localizedseparatorlink_class::getid	296
f_localizerlink_class::getid	296
f_tagged_accumlistedvm_vman_class::getlistatindex	297
f_tri_class::getlocalizer	297
f_polygon_class::getlocalizer	297
f_dividedpolygon_class::getlocalizer	297
f_listedvm_vman_class::getmoments	298
f_vman_class::getnormal	298
f_getvolumemoments::getnormalizedvolumemoments	298
f_planarseparator_class::getnumberofplanes	299
f_polygon_class::getnumberofsimpllicesindecomposition	299
f_dividedpolygon_class::getnumberofsimpllicesindecomposition	299
f_polygon_class::getnumberofvertices	299
f_dividedpolygon_class::getnumberofvertices	300
f_planarseparator_class::getplane	300
f_tri_class::getplaneofexistence	300
f_polygon_class::getplaneofexistence	300
f_dividedpolygon_class::getplaneofexistence	301
f_cutpolygon::getplanepolygonfromreconstruction	301
f_cappeddodecahedron_class::getpt	301
f_cappeddodecahedron_doubles3_class::getpt	301
f_polygon_class::getpt	302
f_polyhedron24_class::getpt	302
f_polyhedron24_doubles3_class::getpt	302
f_dividedpolygon_class::getpt	302
f_cutpolygon::getreconstructionsurfacearea	303
f_polygon_class::getsimplexfromdecomposition	303
f_dividedpolygon_class::getsimplexfromdecomposition	303
f_tagged_accumvm_vm_class::getsize	303
f_tagged_accumvm_sepvm_class::getsize	304
f_listedvm_vman_class::getsize	304
f_bytebuffer_class::getsize	304
f_tagged_accumlistedvm_vman_class::getsize	304
f_tagged_accumvm_sepvm_class::gettagforindex	305
f_tagged_accumvm_vm_class::gettagforindex	305
f_tagged_accumlistedvm_vman_class::gettagforindex	305
f_tri_class::getvertices	305
f_vman_class::getvolume	306
f_sepvm_class::getvolume	306
f_sepvm_doubles3_class::getvolume	306
f_tagged_accumvm_sepvm_class::getvolumeatindex	306
f_tagged_accumvm_vm_class::getvolumeatindex	307
f_tagged_accumvm_sepvm_class::getvolumeattag	307
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, Enable >	307
f_getvolumemoments::getvolumemoments	308
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, NullReconstruction, enable_if_t< IsNullReconstruction< NullReconstruction >::value > > . .	308
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, PlanarSeparator, enable_if_t< is_separated_moments< ReturnType >::value > >	309
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isHalfEdgeCutting< CuttingMethod >::value && IsNotANull< Reconstruction< ReconstructionType >::value &&!IsPlanarSeparator< ReconstructionType >::value &&is_separated_moments< ReturnType >::value> >	310

IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isRecursiveSimplexCutting< CuttingMethod >::value &&!IsNotANullReconstruction< ReconstructionType >::value &&!(IsPlanarSeparator< ReconstructionType >::value &&is_separated_moments< ReturnType >::value)> >	310
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isSimplexCutting< CuttingMethod >::value &&IsNotANullReconstruction< ReconstructionType >::value &&!(IsPlanarSeparator< ReconstructionType >::value &&is_separated_moments< ReturnType >::value)> >	311
f_getvolumemoments::getvolumemoments_setmethod	311
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, ReconstructionType, Enable >	311
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, NullReconstruction, enable_if_t< IsNullReconstruction< NullReconstruction >::value > >	312
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, PlanarSeparator, enable_if_t< isHalfEdgeCutting< CuttingMethod >::value &&is_separated_moments< ReturnType >::value > >	312
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, PlanarSeparator, enable_if_t< isSimplexCutting< CuttingMethod >::value &&is_separated_moments< ReturnType >::value > >	314
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, ReconstructionType, enable_if_t< isHalfEdgeCutting< CuttingMethod >::value &&IsNotANullReconstruction< ReconstructionType >::value &&!(IsPlanarSeparator< ReconstructionType >::value &&is_separated_moments< ReturnType >::value)> >	315
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, ReconstructionType, enable_if_t< isSimplexCutting< CuttingMethod >::value &&IsNotANullReconstruction< ReconstructionType >::value &&!(IsPlanarSeparator< ReconstructionType >::value &&is_separated_moments< ReturnType >::value)> >	315
f_sepvm_class::getvolumeptr	316
f_sepvm_doubles3_class::getvolumeptr	316
f_tagged_accumvm_vm_class::getvolumepratindex	316
f_tagged_accumvm_sepvm_class::getvolumepratindex	316
GraphType	
IRL::ReconstructionLink< ReconstructionType, GraphType >	537
IRL::HalfEdge< VertexType >	317
IRL::HalfEdge< IRL::Vertex >	317
IRL::HalfEdgeCutting	318
IRL::HalfEdgePolytope< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >	321
IRL::HalfEdgePolytope< PtType >	321
IRL::HalfEdgePolygon< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >	318
IRL::HalfEdgePolyhedron< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >	320
IRL::HasACollection< ReturnType >	347
IRL::HasALocalizer< ReconstructionType >	347
IRL::HasANestedType< ReturnType >	348
IRL::HasAReconstructionLink< ReconstructionType >	348
IRL::HasASeparator< ReconstructionType >	348
IRL::HasLocalizer_AND_HasSeparator< ReconstructionType >	349
IRL::HasLocalizer_OR_HasSeparator< ReconstructionType >	349
IRL::HasLocalizer_XOR_HasSeparator< ReconstructionType >	350
HowToContinueDividingVolumeByPlanarReconstruction	
IRL::DivideSimplexByPlanarReconstruction< HowToHandleVolumeCompletelyInternalToPlanarReconstruction, HowToContinueDividingVolumeByPlanarReconstruction >	191
HowToHandleVolumeCompletelyInternalToPlanarReconstruction	

IRL::DivideSimplexByPlanarReconstruction< Reconstruction, HowToHandleVolumeCompletelyInternalToPlanar >	191
IRL::IgnoreVolumeAbovePlane	351
IRL::Illinois< OptimizingClass >	351
f_planarseparator_class::isflipped	407
IRL::IsNotNullReconstruction< ReconstructionType >	410
IRL::IsNotAPlanarSeparator< ReconstructionType >	411
IRL::IsNullReconstruction< ReconstructionType >	411
IRL::IsPlanarSeparator< ReconstructionType >	412
f_geometriccuttinghelpers::isptinternal	412
IRL::IterativeSolverForDistance< CellType, kMaxPlanes >	418
iterator	
IRL::ConstIteratorThroughBracketOperator< ContainerType >	177
IRL::IteratorThroughBracketOperator< ContainerType >	422
IRL::JoinedReconstructions< CurrentReconstruction, NextReconstruction >	423
IRL::JoinedReconstructions< CurrentReconstruction, NullReconstruction >	424
IRL::JoinedReconstructionsToGroup< CurrentReconstruction, GroupReconstruction >	425
IRL::KMeans	425
IRL::LevenbergMarquardt< OptimizingClass, kRows, kColumns >	426
IRL::LevenbergMarquardt< OptimizingClass, -1, kColumns >	429
IRL::LinearInterpolation_Functor	430
f_listedvmm_vman_class::listedvmm_vman_type	430
IRL::ListedVolumeMoments< VolumeMomentsType >	431
IRL::ListedVolumeMoments< IRL::VolumeMomentsAndNormal >	431
f_localizedseparatorlink_class::localizedseparatorlink_type	434
f_localizerlink_class::localizerlink_type	435
IRL::LVIRACommon< CellType, kColumns >	442
IRL::LVIRACommon< CellType, LVIRA_2D_columns >	442
IRL::LVIRA_2D< CellType >	436
IRL::LVIRACommon< CellType, LVIRA_3D_columns >	442
IRL::LVIRA_3D< CellType >	439
IRL::LVIRANeighborhood< CellType >	447
IRL::LVIRANeighborhood< IRL::StoredRectangularCuboid >	447
f_lviranighborhood_rectangularcuboid_class::lviranighborhood_rectangularcuboid_type	450
LVIRAType	
IRL::LVIRADebug< LVIRAType >	446
MaskedType	
IRL::MaskStripper< MaskedType >	451
IRL::MathVector< kNumberOfElements >	452
IRL::MathVector< 3 >	452
IRL::MOFCommon< CellType >	456
IRL::MOF_2D< CellType >	453
IRL::MOF_3D< CellType >	455
MOFType	
IRL::MOFDebug< MOFType >	459
IRL::MomentCalculationType< VertexType, ProxyType, Enable >	461
IRL::MomentCalculationType< VertexType, ProxyTet< DecomposedPolyhedronVertexStorage< VertexType >>>	461
IRL::SegmentedDecomposedPolytope< DecomposedPolyhedronVertexStorage< VertexType >, VertexType, ProxyTet< DecomposedPolyhedronVertexStorage< VertexType >>>	553
IRL::SegmentedDecomposedPolyhedron< VertexType >	552
IRL::MomentCalculationType< VertexType, ProxyTri< DecomposedPolygonVertexStorage< VertexType >>>	461
IRL::SegmentedDecomposedPolytope< DecomposedPolygonVertexStorage< VertexType >, VertexType, ProxyTri< DecomposedPolygonVertexStorage< VertexType >>>	553
IRL::SegmentedDecomposedPolygon< VertexType >	551
IRL::MomentCalculationType< VertexType, ProxyType >	461

IRL::SegmentedDecomposedPolytope< VertexStorageType, VertexType, ProxyType >	553
f_tagged_accumvm_sepvm_class::multiplybyvolume	463
f_tagged_accumvm_vm_class::multiplybyvolume	464
f_vman_class::multiplybyvolume	464
f_sepvm_class::multiplybyvolume	464
f_sepvm_doubles3_class::multiplybyvolume	464
f_objectallocationserver_localizedseparatorlink_class::new	465
f_elviraneighborhood_class::new	465
f_objectallocationserver_planarlocalizer_class::new	465
f_planarlocalizer_class::new	465
f_tet_class::new	466
f_polygon_class::new	466
f_dodecahedron_class::new	466
f_planarseparator_class::new	466
f_rectangularcuboid_class::new	467
f_vman_class::new	467
f_tri_class::new	467
f_cappeddodecahedron_doubles3_class::new	467
f_tagged_accumvm_vm_class::new	468
f_tagged_accumlistedvm_vman_class::new	468
f_dividedpolygon_class::new	468
f_polyhedron24_class::new	468
f_polyhedron24_doubles3_class::new	469
f_listedvm_vman_class::new	469
f_bytebuffer_class::new	469
f_objectallocationserver_planarseparator_class::new	469
f_r2pneighborhood_rectangularcuboid_class::new	470
f_localizedseparatorlink_class::new	470
f_localizerlink_class::new	470
f_objectallocationserver_localizerlink_class::new	470
f_sepvm_class::new	471
f_sepvm_doubles3_class::new	471
f_lviraneighborhood_rectangularcuboid_class::new	471
f_tagged_accumvm_sepvm_class::new	471
f_cappeddodecahedron_class::new	472
IRL::Normal	472
f_tagged_accumvm_sepvm_class::normalizebyvolume	474
f_tagged_accumvm_vm_class::normalizebyvolume	474
f_sepvm_class::normalizebyvolume	475
f_sepvm_doubles3_class::normalizebyvolume	475
f_vman_class::normalizebyvolume	475
IRL::NullReconstruction	475
IRL::ObjectAllocationServer< ObjectType >	476
IRL::ObjectAllocationServer< IRL::PlanarLocalizer >	476
IRL::ObjectAllocationServer< IRL::PlanarSeparator >	476
IRL::ObjectAllocationServer< IRL::ReconstructionLink >	476
f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_type .	476
f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_type	477
f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer_type	477
f_objectallocationserver_planarseparator_class::objectallocationserver_planarseparator_type	478
IRL::PartitionByNormal< MomentsContainerType >	478
IRL::PassToNestedType	480
IRL::PathGraphNode< NodeType >	480
IRL::PlanarLocalizer	482
f_planarlocalizer_class::planarlocalizer_type	484
IRL::PlanarReconstruction< kStackPlanes >	485
IRL::PlanarReconstruction< global_constants::MAX_PLANAR_LOCALIZER_PLANES >	485
IRL::PlanarSeparator	487

f_planarseparator_class::planarseparator_type	489
IRL::PlanarSeparatorPathGroup	490
IRL::Plane	491
PlaneOfExistenceType	
IRL::PolygonConstructors< Derived, VertexType, StorageType, PlaneOfExistenceType >	493
IRL::ExpandableDividedPolygon< VertexType >	204
IRL::ExpandablePolygon< VertexType >	205
IRL::StoredTri< VertexType >	589
IRL::ProxyPolygonConstructors< Derived, GeometryType, kNumberOfVertices, PlaneOfExistence← Type >	505
IRL::ProxyTri< GeometryType >	507
f_polygon_class::polygon_type	492
IRL::PolygonBase< VertexType >	493
IRL::PolygonMomentsCalculationCommon< Derived, VertexType, SimplexType >	495
IRL::PolygonMomentsCalculation< Derived, VertexType, SimplexType >	494
IRL::GeneralPolygon< Derived, VertexType, SimplexType >	279
IRL::PolygonMomentsCalculationCommon< Derived, VertexType, ProxyTri< Derived > >	495
IRL::PolygonMomentsCalculation< Derived, VertexType, ProxyTri< Derived > >	494
IRL::GeneralPolygon< Derived, VertexType, ProxyTri< Derived > >	279
IRL::PolygonSpecialization< Derived, VertexType >	496
IRL::TriSpecialization< Derived, VertexType >	600
IRL::PolygonMomentsCalculationCommon< Derived, VertexType, ProxyTri< MaskStripper< Derived > > >	495
IRL::PolygonMomentsCalculation< Derived, VertexType, ProxyTri< MaskStripper< Derived > > >	494
IRL::GeneralPolygon< Derived, VertexType, ProxyTri< MaskStripper< Derived > > >	279
IRL::DividedPolygonSpecialization< Derived, VertexType >	189
IRL::PolygonMomentsCalculationCommon< ExpandableDividedPolygon< VertexType >, VertexType, ProxyTri< MaskStripper< ExpandableDividedPolygon< VertexType > >> >	495
IRL::PolygonMomentsCalculation< ExpandableDividedPolygon< VertexType >, VertexType, Proxy← Tri< MaskStripper< ExpandableDividedPolygon< VertexType > >> >	494
IRL::GeneralPolygon< ExpandableDividedPolygon< VertexType >, VertexType, ProxyTri< MaskStripper< ExpandableDividedPolygon< VertexType > >> >	279
IRL::DividedPolygonSpecialization< ExpandableDividedPolygon< VertexType >, VertexType >	189
IRL::ExpandableDividedPolygon< VertexType >	204
IRL::PolygonMomentsCalculationCommon< ExpandablePolygon< VertexType >, VertexType, ProxyTri< ExpandablePolygon< VertexType > >>	495
IRL::PolygonMomentsCalculation< ExpandablePolygon< VertexType >, VertexType, ProxyTri< ExpandablePolygon< VertexType > >>	494
IRL::GeneralPolygon< ExpandablePolygon< VertexType >, VertexType, ProxyTri< ExpandablePolygon< VertexType > >>	279
IRL::PolygonSpecialization< ExpandablePolygon< VertexType >, VertexType >	496
IRL::ExpandablePolygon< VertexType >	205
IRL::PolygonMomentsCalculationCommon< ProxyTri< GeometryType >, GeometryType::pt_type, ProxyTri< ProxyTri< GeometryType > >>	495
IRL::PolygonMomentsCalculation< ProxyTri< GeometryType >, GeometryType::pt_type, ProxyTri< ProxyTri< GeometryType > >>	494
IRL::GeneralPolygon< ProxyTri< GeometryType >, GeometryType::pt_type, ProxyTri< ProxyTri< GeometryType > >>	279
IRL::TriSpecialization< ProxyTri< GeometryType >, GeometryType::pt_type >	600
IRL::ProxyTri< GeometryType >	507
IRL::PolygonMomentsCalculationCommon< SegmentedDecomposedPolygon< VertexType >, Vertex← Type, ProxyType >	495
IRL::PolygonMomentsCalculation< SegmentedDecomposedPolygon< VertexType >, VertexType, ProxyType >	494
IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polygon< ProxyType >← ::value >>	461

IRL::PolygonMomentsCalculationCommon< StoredTri< VertexType >, VertexType, ProxyTri< StoredTri< VertexType > >>	495
IRL::PolygonMomentsCalculation< StoredTri< VertexType >, VertexType, ProxyTri< StoredTri< VertexType > >>	494
IRL::GeneralPolygon< StoredTri< VertexType >, VertexType, ProxyTri< StoredTri< VertexType > >	279
IRL::TriSpecialization< StoredTri< VertexType >, VertexType >	600
IRL::StoredTri< VertexType >	589
IRL::Polyhedron< VertexType, kMaxNumberOfPts >	499
f_polyhedron24_doubles3_class::polyhedron24_doubles3_type	499
f_polyhedron24_class::polyhedron24_type	500
IRL::PolyhedronMomentsCalculationCommon< Derived, VertexType, SimplexType >	503
IRL::PolyhedronMomentsCalculation< Derived, VertexType, SimplexType >	501
IRL::GeneralPolyhedron< Derived, VertexType, SimplexType >	281
IRL::PolyhedronMomentsCalculationCommon< Derived, PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, SimplexType >	503
IRL::PolyhedronMomentsCalculation< Derived, PtWithDoublesStatelessFunctor< FunctorType, k← ArrayLength >, SimplexType >	502
IRL::PolyhedronMomentsCalculationCommon< Derived, VertexType, ProxyTet< Derived > >	503
IRL::PolyhedronMomentsCalculation< Derived, VertexType, ProxyTet< Derived > >	501
IRL::GeneralPolyhedron< Derived, VertexType, ProxyTet< Derived > >	281
IRL::CappedDodecahedronSpecialization< Derived, VertexType >	167
IRL::ConcaveBoxSpecialization< Derived, VertexType >	176
IRL::DodecahedronSpecialization< Derived, VertexType >	192
IRL::HexahedronSpecialization< Derived, VertexType >	350
IRL::RectangularCuboidSpecialization< Derived, VertexType >	542
IRL::Polyhedron24Specialization< Derived, VertexType >	500
IRL::TetSpecialization< Derived, VertexType >	598
IRL::PolyhedronMomentsCalculationCommon< ProxyTet< GeometryType >, GeometryType::pt_type, ProxyTet< ProxyTet< GeometryType > >>	503
IRL::PolyhedronMomentsCalculation< ProxyTet< GeometryType >, GeometryType::pt_type, Proxy← Tet< ProxyTet< GeometryType > >>	501
IRL::GeneralPolyhedron< ProxyTet< GeometryType >, GeometryType::pt_type, ProxyTet< ProxyTet< GeometryType > >>	281
IRL::TetSpecialization< ProxyTet< GeometryType >, GeometryType::pt_type >	598
IRL::ProxyTet< GeometryType >	506
IRL::PolyhedronMomentsCalculationCommon< SegmentedDecomposedPolyhedron< VertexType >, VertexType, ProxyType >	503
IRL::PolyhedronMomentsCalculation< SegmentedDecomposedPolyhedron< VertexType >, Vertex← Type, ProxyType >	501
IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polyhedron< ProxyType >::value > >	461
IRL::PolyhedronMomentsCalculationCommon< StoredCappedDodecahedron< IRL::PtWithDoubles← StatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWithDoublesStateless← Functor< IRL::LinearInterpolation_Functor, 3 >, ProxyTet< StoredCappedDodecahedron< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >> >	503
IRL::PolyhedronMomentsCalculation< StoredCappedDodecahedron< IRL::PtWithDoubles← StatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWithDoublesStateless← Functor< IRL::LinearInterpolation_Functor, 3 >, ProxyTet< StoredCappedDodecahedron< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >> >	501
IRL::GeneralPolyhedron< StoredCappedDodecahedron< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWithDoublesStatelessFunctor< IRL::← LinearInterpolation_Functor, 3 >, ProxyTet< StoredCappedDodecahedron< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >> >	281
IRL::CappedDodecahedronSpecialization< StoredCappedDodecahedron< IRL::PtWith← DoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWith← DoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >	167

IRL::StoredCappedDodecahedron< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >	584
IRL::PolyhedronMomentsCalculationCommon< StoredCappedDodecahedron< VertexType >, VertexType, ProxyTet< StoredCappedDodecahedron< VertexType > >>	503
IRL::PolyhedronMomentsCalculation< StoredCappedDodecahedron< VertexType >, VertexType, ProxyTet< StoredCappedDodecahedron< VertexType > >>	501
IRL::GeneralPolyhedron< StoredCappedDodecahedron< VertexType >, VertexType, ProxyTet< StoredCappedDodecahedron< VertexType > >>	281
IRL::CappedDodecahedronSpecialization< StoredCappedDodecahedron< VertexType >, VertexType >	167
IRL::StoredCappedDodecahedron< VertexType >	584
IRL::PolyhedronMomentsCalculationCommon< StoredConcaveBox< VertexType >, VertexType, ProxyTet< StoredConcaveBox< VertexType > >>	503
IRL::PolyhedronMomentsCalculation< StoredConcaveBox< VertexType >, VertexType, ProxyTet< StoredConcaveBox< VertexType > >>	501
IRL::GeneralPolyhedron< StoredConcaveBox< VertexType >, VertexType, ProxyTet< StoredConcaveBox< VertexType > >>	281
IRL::ConcaveBoxSpecialization< StoredConcaveBox< VertexType >, VertexType >	176
IRL::StoredConcaveBox< VertexType >	585
IRL::PolyhedronMomentsCalculationCommon< StoredDodecahedron< VertexType >, VertexType, ProxyTet< StoredDodecahedron< VertexType > >>	503
IRL::PolyhedronMomentsCalculation< StoredDodecahedron< VertexType >, VertexType, ProxyTet< StoredDodecahedron< VertexType > >>	501
IRL::GeneralPolyhedron< StoredDodecahedron< VertexType >, VertexType, ProxyTet< StoredDodecahedron< VertexType > >>	281
IRL::DodecahedronSpecialization< StoredDodecahedron< VertexType >, VertexType >	192
IRL::StoredDodecahedron< VertexType >	585
IRL::PolyhedronMomentsCalculationCommon< StoredHexahedron< VertexType >, VertexType, ProxyTet< StoredHexahedron< VertexType > >>	503
IRL::PolyhedronMomentsCalculation< StoredHexahedron< VertexType >, VertexType, ProxyTet< StoredHexahedron< VertexType > >>	501
IRL::GeneralPolyhedron< StoredHexahedron< VertexType >, VertexType, ProxyTet< StoredHexahedron< VertexType > >>	281
IRL::HexahedronSpecialization< StoredHexahedron< VertexType >, VertexType >	350
IRL::StoredHexahedron< VertexType >	586
IRL::PolyhedronMomentsCalculationCommon< StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 >, ProxyTet< StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >>>	503
IRL::PolyhedronMomentsCalculation< StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 >, ProxyTet< StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >>>	501
IRL::GeneralPolyhedron< StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 >, ProxyTet< StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >>>	281
IRL::Polyhedron24Specialization< StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >	500
IRL::StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >	587
IRL::PolyhedronMomentsCalculationCommon< StoredPolyhedron24< VertexType >, VertexType, ProxyTet< StoredPolyhedron24< VertexType > >>	503
IRL::PolyhedronMomentsCalculation< StoredPolyhedron24< VertexType >, VertexType, ProxyTet< StoredPolyhedron24< VertexType > >>	501

IRL::GeneralPolyhedron< StoredPolyhedron24< VertexType >, VertexType, ProxyTet< StoredPolyhedron24< VertexType > >>	281
IRL::Polyhedron24Specialization< StoredPolyhedron24< VertexType >, VertexType >	500
IRL::StoredPolyhedron24< VertexType >	587
IRL::PolyhedronMomentsCalculationCommon< StoredRectangularCuboid< VertexType >, VertexType, ProxyTet< StoredRectangularCuboid< VertexType > >>	503
IRL::PolyhedronMomentsCalculation< StoredRectangularCuboid< VertexType >, VertexType, ProxyTet< StoredRectangularCuboid< VertexType > >>	501
IRL::GeneralPolyhedron< StoredRectangularCuboid< VertexType >, VertexType, ProxyTet< StoredRectangularCuboid< VertexType > >>	281
IRL::HexahedronSpecialization< StoredRectangularCuboid< VertexType >, VertexType >	350
IRL::RectangularCuboidSpecialization< StoredRectangularCuboid< VertexType >, VertexType >	542
IRL::StoredRectangularCuboid< VertexType >	588
IRL::PolyhedronMomentsCalculationCommon< StoredTet< VertexType >, VertexType, ProxyTet< StoredTet< VertexType > >>	503
IRL::PolyhedronMomentsCalculation< StoredTet< VertexType >, VertexType, ProxyTet< StoredTet< VertexType > >>	501
IRL::GeneralPolyhedron< StoredTet< VertexType >, VertexType, ProxyTet< StoredTet< VertexType > >>	281
IRL::TetSpecialization< StoredTet< VertexType >, VertexType >	598
IRL::StoredTet< VertexType >	588
f_dividedpolygon_class::printtoscreen	504
f_planarlocalizer_class::printtoscreen	504
f_planarseparator_class::printtoscreen	504
f_polygon_class::printtoscreen	505
IRL::ProxyVertexAccess< Derived, GeometryType, kNumberOfVertices >	508
IRL::ProxyPolygonConstructors< Derived, GeometryType, kNumberOfVertices, PlaneOfExistence< Type >	505
IRL::ProxyVertexAccess< ProxyTet< GeometryType >, GeometryType, 4 >	508
IRL::ProxyTet< GeometryType >	506
IRL::PtList< VertexType, kMaxNumberOfPts >	513
IRL::PtList< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 >, kNumberOfVertices >	513
IRL::PtList< VertexType, kNumberOfVertices >	513
IRL::PtWithDataCommon< Derived, AttachedDataType >	514
IRL::PtWithDataCommon< PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, std::array< double, kArrayLength > >	514
IRL::PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >	515
IRL::R2PCommon< CellType, kColumns >	527
IRL::R2PCommon< CellType, R2P_2D1P_columns >	527
IRL::R2P_2D1P< CellType >	516
IRL::R2PCommon< CellType, R2P_2D2P_columns >	527
IRL::R2P_2D2P< CellType >	519
IRL::R2PCommon< CellType, R2P_3D1P_columns >	527
IRL::R2P_3D1P< CellType >	521
IRL::R2PCommon< CellType, R2P_3D2P_columns >	527
IRL::R2P_3D2P< CellType >	524
IRL::R2PNeighborhood< CellType >	534
IRL::R2PNeighborhood< IRL::StoredRectangularCuboid >	534
f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_type	536
R2PType	
IRL::R2PDebug< R2PType >	532
short_alloc::short_alloc< T, N, Align >::rebind< _Up >	537
ReconstructionType	

IRL::ReconstructionLink< ReconstructionType, GraphType >	537
f_reconstructioninterface::reconstructionwithadvedtednormals	538
f_reconstructioninterface::reconstructionwithadvedtednormalsdebug	539
f_reconstructioninterface::reconstructionwithlvira2d	539
f_reconstructioninterface::reconstructionwithlvira3d	539
f_reconstructioninterface::reconstructionwithmof2d	539
f_reconstructioninterface::reconstructionwithmof3d	540
f_reconstructioninterface::reconstructionwithr2p2d	540
f_reconstructioninterface::reconstructionwithr2p2ddebug	540
f_reconstructioninterface::reconstructionwithr2p3d	540
f_reconstructioninterface::reconstructionwithr2p3ddebug	541
f_rectangularcuboid_class::rectangularcuboid_type	541
IRL::RectangularCuboidBase< VertexType >	541
IRL::RectangularCuboidCommon< Derived, VertexType >	542
IRL::RecursiveSimplexCutting	543
IRL::ReferenceFrame	543
IRL::ReferredToPlaneOFExistence	544
f_bytebuffer_class::resetbufferpointer	544
f_dividedpolygon_class::resetcentroid	545
f_tri_class::reverseptordering	547
f_dividedpolygon_class::reverseptordering	547
f_polygon_class::reverseptordering	547
IRL::Secant< OptimizingClass >	548
IRL::SegmentedHalfEdgePolygonSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices >	556
IRL::SegmentedHalfEdgePolygonSpecificPt< VertexType::vertex_location_type::pt_type, FaceType, VertexType, kMaxFaces, kMaxVertices >	556
IRL::SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices >	554
IRL::SegmentedHalfEdgePolytope< FaceType, VertexType, kMaxFaces, kMaxVertices >	559
IRL::SegmentedHalfEdgePolytope< FaceType, VertexType >	559
IRL::SegmentedHalfEdgePolygonCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >	555
IRL::SegmentedHalfEdgePolygonSpecificPt< Pt, FaceType, VertexType, kMaxFaces, kMaxVertices, kMaxVertices >	556
IRL::SegmentedHalfEdgePolyhedronCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >	557
IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices, kMaxVertices >	558
IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, FaceType, VertexType, kMaxFaces, kMaxVertices >	559
IRL::SegmentedHalfEdgePolyhedronSpecificPt< VertexType::vertex_location_type::pt_type, FaceType, VertexType, kMaxFaces, kMaxVertices >	558
IRL::SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices >	556
IRL::SeparatedMoments< MomentsType >	562
IRL::SeparatedMoments< IRL::VolumeMoments >	562
IRL::SeparatedMoments< IRL::VolumeMomentsAndDoubles< 3 > >	562
IRL::SeparatedMomentsCommon< Derived, MomentsType >	564
IRL::SeparatedMomentsCommon< SeparatedMoments< VolumeMoments >, VolumeMoments >	564
IRL::SeparatedMoments< VolumeMoments >	562
IRL::SeparatedMomentsCommon< SeparatedMoments< VolumeMomentsAndDoubles< kArrayLength > >, VolumeMomentsAndDoubles< kArrayLength > >	564
IRL::SeparatedMoments< VolumeMomentsAndDoubles< kArrayLength > >	563
f_sepvm_doubles3_class::sepvm_doubles3_type	568
f_sepvm_class::sepvm_type	568
f_serializer::serializeandpack	569
f_lviranighborhood_rectangularcuboid_class::setcenterofstencil	569
f_r2pneighborhood_rectangularcuboid_class::setcenterofstencil	569
f_polyhedron24_doubles3_class::setdata	570
f_cappeddodecahedron_doubles3_class::setdata	570
f_volumefractionmatching::setdistancetomatchvolumefraction	570

f_localizedseparatorlink_class::setedgeconnectivity	570
f_localizerlink_class::setedgeconnectivity	571
f_localizedseparatorlink_class::setedgeconnectivitynull	571
f_localizerlink_class::setedgeconnectivitynull	571
f_planarlocalizer_class::setfromrectangularcuboid	571
f_localizerlink_class::setid	572
f_localizedseparatorlink_class::setid	572
f_elviraneighborhood_class::setmember	572
f_r2pneighborhood_rectangularcuboid_class::setmember	572
f_lviraneighborhood_rectangularcuboid_class::setmember	573
f_planarseparator_class::setnumberofplanes	573
f_planarlocalizer_class::setnumberofplanes	573
f_planarlocalizer_class::setplane	573
f_planarseparator_class::setplane	574
f_tri_class::setplaneofexistence	574
f_polygon_class::setplaneofexistence	574
f_dividedpolygon_class::setplaneofexistence	574
f_polyhedron24_doubles3_class::setpt	575
f_cappeddodecahedron_doubles3_class::setpt	575
f_polyhedron24_class::setpt	575
f_elviraneighborhood_class::setsize	575
f_lviraneighborhood_rectangularcuboid_class::setsize	576
f_r2pneighborhood_rectangularcuboid_class::setsize	576
f_bytebuffer_class::setsize	576
f_r2pneighborhood_rectangularcuboid_class::setsurfacearea	576
IRL::ShareVolumeAbovePlaneWithLinkedNeighbor	577
short_alloc::short_alloc< T, N, Align >	577
IRL::SimplexCutting	578
IRL::SimplexWrapper< SimplexType, Enable >	578
IRL::SimplexWrapper< SimplexType, enable_if_t< is_tet< SimplexType >::value > >	578
IRL::SimplexWrapper< SimplexType, enable_if_t< is_tri< SimplexType >::value > >	580
IRL::SmallVector< T, BuffSizelnElements >	580
IRL::SmallVector< double, kMaxPlanes >	580
IRL::SmallVector< FaceType *, kMaxFaces >	580
IRL::SmallVector< FaceType *, segmented_half_edge_polytope::default_sizes::segemented_max_faces >	580
IRL::SmallVector< IRL::Plane, kStackPlanes >	580
IRL::SmallVector< ProxyTet< DecomposedPolyhedronVertexStorage< VertexType > >, 40 >	580
IRL::SmallVector< ProxyTri< DecomposedPolygonVertexStorage< VertexType > >, 40 >	580
IRL::SmallVector< ProxyType, 40 >	580
IRL::SmallVector< uint8_t, 40 >	580
IRL::SmallVector< UnsignedIndex_t, 40 >	580
IRL::SmallVector< VertexType *, kMaxVertices >	580
IRL::SmallVector< VertexType *, segmented_half_edge_polytope::default_sizes::segmented_max_<>vertices >	580
IRL::SmallVector< VertexType, 10 >	580
IRL::StackVector< ObjectType, kMaxSize >	582
IRL::StolenGraph< NodeType, StolenGraphType >	583
IRL::StolenGraph< LocalizerLinkFromLocalizedSeparatorLink, LocalizedSeparatorLink >	583
IRL::LocalizerLinkFromLocalizedSeparatorLink	435
IRL::StolenGraph< MaskedLocalizedSeparatorLink, LocalizedSeparatorLink >	583
IRL::MaskedLocalizedSeparatorLink	450
IRL::RestrictedLocalizerLinkFromLocalizedSeparatorLink	545
IRL::StolenGraph< RestrictedMaskedLocalizedSeparatorLink, LocalizedSeparatorLink >	583
IRL::RestrictedMaskedLocalizedSeparatorLink	546
StorageType	
IRL::PolygonConstructors< Derived, VertexType, StorageType, PlaneOfExistenceType >	493
IRL::StoredPlaneOfExistence	586

IRL::StoredVertexAccess< Derived, VertexType, kNumberOfVertices >	590
IRL::StoredVertexAccess< StoredCappedDodecahedron< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 >, 9 >	590
IRL::StoredCappedDodecahedron< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >	584
IRL::StoredVertexAccess< StoredCappedDodecahedron< VertexType >, VertexType, 9 >	590
IRL::StoredCappedDodecahedron< VertexType >	584
IRL::StoredVertexAccess< StoredConcaveBox< VertexType >, VertexType, 14 >	590
IRL::StoredConcaveBox< VertexType >	585
IRL::StoredVertexAccess< StoredDodecahedron< VertexType >, VertexType, 8 >	590
IRL::StoredDodecahedron< VertexType >	585
IRL::StoredVertexAccess< StoredHexahedron< VertexType >, VertexType, 8 >	590
IRL::StoredHexahedron< VertexType >	586
IRL::StoredVertexAccess< StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >, IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 >, 14 >	590
IRL::StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >	587
IRL::StoredVertexAccess< StoredPolyhedron24< VertexType >, VertexType, 14 >	590
IRL::StoredPolyhedron24< VertexType >	587
IRL::StoredVertexAccess< StoredRectangularCuboid< VertexType >, VertexType, 8 >	590
IRL::StoredRectangularCuboid< VertexType >	588
IRL::StoredVertexAccess< StoredTet< VertexType >, VertexType, 4 >	590
IRL::StoredTet< VertexType >	588
IRL::StrongWolfeConditionLineSearch	591
f_tagged_accumlistedvmm_vman_class::tagged_accumlistedvmm_vman_type	592
f_tagged_accumvmm_sepvm_class::tagged_accumvmm_sepvm_type	593
f_tagged_accumvmm_vmm_class::tagged_accumvmm_vmm_type	594
IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType >	595
IRL::TaggedAccumulatedVolumeMoments< IRL::SeparatedMoments< IRL::VolumeMoments > >	595
IRL::TaggedAccumulatedVolumeMoments< IRL::VolumeMoments >	595
IRL::TaggedAccumulatedVolumeMoments< ListedVolumeMoments< IRL::VolumeMomentsAndNormal > >	595
IRL::TaggedAccumulatedListedVolumeMoments< IRL::VolumeMomentsAndNormal >	594
IRL::TaggedAccumulatedVolumeMoments< ListedVolumeMoments< VolumeMomentsType > >	595
IRL::TaggedAccumulatedListedVolumeMoments< VolumeMomentsType >	594
IRL::TaggedAccumulatedVolumeMoments< MomentsType >	595
IRL::MultiMaterialAccumulatedVolumeMoments< MomentsType >	462
IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType >::TaggedMoments	597
f_tet_class::tet_type	597
f_tri_class::tri_type	599
true_type	
IRL::has_localizer< LocalizedSeparator >	324
IRL::has_localizer< LocalizedSeparatorGroup >	325
IRL::has_localizer< LocalizedSeparatorGroupLink >	326
IRL::has_localizer< LocalizedSeparatorLink >	327
IRL::has_localizer< LocalizerLink >	328
IRL::has_localizer< LocalizerLinkFromLocalizedSeparatorLink >	329
IRL::has_localizer< MaskedLocalizedSeparatorLink >	330
IRL::has_localizer< PlanarLocalizer >	331
IRL::has_localizer< RestrictedLocalizerLinkFromLocalizedSeparatorLink >	332
IRL::has_localizer< RestrictedMaskedLocalizedSeparatorLink >	333
IRL::has_null_reconstruction< NullReconstruction >	336
IRL::has_separator< LocalizedSeparator >	339

IRL::has_separator< LocalizedSeparatorGroup >	340
IRL::has_separator< LocalizedSeparatorGroupLink >	341
IRL::has_separator< LocalizedSeparatorLink >	342
IRL::has_separator< MaskedLocalizedSeparatorLink >	343
IRL::has_separator< PlanarSeparator >	344
IRL::has_separator< PlanarSeparatorPath >	345
IRL::has_separator< RestrictedMaskedLocalizedSeparatorLink >	346
IRL::is_moments_collection< AccumulatedListedVolumeMoments< VolumeMomentsType >>	355
IRL::is_moments_collection< AccumulatedVolumeMoments< VolumeMomentsType >>	356
IRL::is_moments_collection< ListedVolumeMoments< VolumeMomentsType >>	358
IRL::is_moments_collection< TaggedAccumulatedListedVolumeMoments< VolumeMomentsType >>	359
IRL::is_moments_collection< TaggedAccumulatedVolumeMoments< VolumeMomentsType >>	360
IRL::is_nested_moments< AccumulateWrapper< NestedType >>	362
IRL::is_planar_separator< PlanarSeparator >	366
IRL::is_polygon< ExpandableDividedPolygon< VertexType >>	369
IRL::is_polygon< ExpandablePolygon< VertexType >>	370
IRL::is_polygon< ProxyTri< GeometryType >>	371
IRL::is_polygon< SegmentedDecomposedPolygon< VertexType >>	372
IRL::is_polygon< SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices > >	373
IRL::is_polygon< StoredTri< VertexType >>	374
IRL::is_polyhedron< ProxyTet< GeometryType >>	377
IRL::is_polyhedron< SegmentedDecomposedPolyhedron< VertexType >>	378
IRL::is_polyhedron< SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices >>	379
IRL::is_polyhedron< StoredCappedDodecahedron< VertexType >>	380
IRL::is_polyhedron< StoredConcaveBox< VertexType >>	381
IRL::is_polyhedron< StoredDodecahedron< VertexType >>	382
IRL::is_polyhedron< StoredHexahedron< VertexType >>	383
IRL::is_polyhedron< StoredPolyhedron24< VertexType >>	384
IRL::is_polyhedron< StoredRectangularCuboid< VertexType >>	385
IRL::is_polyhedron< StoredTet< VertexType >>	386
IRL::is_reconstruction_link< LocalizedSeparatorGroupLink >	389
IRL::is_reconstruction_link< LocalizedSeparatorLink >	390
IRL::is_reconstruction_link< LocalizerLink >	391
IRL::is_reconstruction_link< LocalizerLinkFromLocalizedSeparatorLink >	392
IRL::is_reconstruction_link< MaskedLocalizedSeparatorLink >	393
IRL::is_reconstruction_link< PlanarSeparatorPath >	394
IRL::is_reconstruction_link< RestrictedLocalizerLinkFromLocalizedSeparatorLink >	395
IRL::is_reconstruction_link< RestrictedMaskedLocalizedSeparatorLink >	396
IRL::is_separated_moments< SeparatedMoments< MomentsType >>	399
IRL::is_tet< ProxyTet< GeometryType >>	402
IRL::is_tet< StoredTet< VertexType >>	403
IRL::is_tri< ProxyTri< GeometryType >>	406
IRL::is_tri< StoredTri< VertexType >>	407
IRL::isHalfEdgeCutting< HalfEdgeCutting >	410
IRL::isRecursiveSimplexCutting< RecursiveSimplexCutting >	414
IRL::isSimplexCutting< SimplexCutting >	417
IRL::UnDirectedGraphNode< NodeType >	602
IRL::UnitQuaternion	603
f_serializer::unpackandstore	604
IRL::Vertex< VertexLocationType >	606
IRL::VertexList< VertexStorage, VertexType >	607
IRL::VertexList< DecomposedPolygonVertexStorage< VertexType >, VertexType >	607
IRL::VertexList< DecomposedPolyhedronVertexStorage< VertexType >, VertexType >	607
IRL::VertexList< VertexStorageType, VertexType >	607
IRL::VertexLocation< PtType >	608
f_vman_class::vman_type	608

IRL::Volume	609
IRL::Volume2D_Functor	610
IRL::Volume3D_Functor	611
IRL::VolumeMoments	611
IRL::VolumeMoments2D_Functor	614
IRL::VolumeMoments3D_Functor	615
IRL::VolumeMomentsAndDoubles< kArrayLength >	616
IRL::VolumeMomentsAndDoubles3D_Functor< kArrayLength >	618
IRL::VolumeMomentsAndNormal	619
IRL::VolumeMomentsAndNormal2D_Functor	621
WrappedType	
IRL::AccumulateWrapper< WrappedType >	123
f_listedvm_vman_class::zeronormalcomponent	622
f_dividedpolygon_class::zeropolygon	622
f_polygon_class::zeropolygon	622

Chapter 3

Class Index

3.1 Class List

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

IRL::AccumulatedListedVolumeMoments< VolumeMomentsType >	119
IRL::AccumulatedVolumeMoments< VolumeMomentsType >	
Self-expanding vector of VolumeMoments	121
IRL::AccumulateIntoCollection	122
IRL::AccumulateIntoScalar	123
IRL::AccumulateWrapper< WrappedType >	
This is essentially a class that delays the evaluation of Tagged moments during generic_cutting with a ReconstructionLink for an eventual collection. As an example, this class needs to be used when cutting by a LocalizedSeparatorPath for tagged moments, since the first cutting operation (by the Localizer), is not what will dictate the tagging of the moments, but instead the second (by the PlanarSeparatorPath) does. The use of the tagged moments that are going to be returned need to be delayed and accumulated. We would then return them as auto moments = getVolumeMoments<AccumulateWrapper<TaggedAccumulatedVolumeMoments<VolumeMoments>>> (a_volume_to_cut, a_LocalizedSeparatorPath);	123
IRL::Accumulator< ObjectType >	
This class is a SelfExpandingCollection that has operator += call the += operator for each member in its collection	124
f_lviraneighborhood_rectangularcuboid_class::addmember	125
f_r2neighborhood_rectangularcuboid_class::addmember	126
f_planarseparator_class::addplane	126
f_planarlocalizer_class::addplane	126
f_cappeddodecahedron_doubles3_class::adjustcaptomatchvolume	126
f_cappeddodecahedron_class::adjustcaptomatchvolume	127
f_polyhedron24_class::adjustcaptomatchvolume	127
f_polyhedron24_doubles3_class::adjustcaptomatchvolume	127
IRL::AdvectionPlaneReconstruction	127
IRL::AdvectionPlaneReconstructionDebug	128
f_listedvmm_vman_class::append	129
f_tagged_accumlistedvmm_vman_class::append	129
short_alloc::arena< N, alignment >	129
IRL::BFGS< OptimizingClass, kParameters >	
Implementation of BFGS method according to the book Numerical Optimization by Jorge Nocedal and Stephen Wright, 2006	130
IRL::Bisection< OptimizingClass >	
A templated class driver that performs bisection	131

IRL::BlockObjectAllocation< ObjectType >	133
IRL::BrentsMethod< OptimizingClass >	
A templated class driver that performs Brent's Method	134
IRL::ByteBuffer	136
f_bytebuffer_class::bytebuffer_type	137
f_bytebuffer_class::c_bytebuffer	137
c_ByteBuffer	138
c_CappedDodecahedron	138
f_cappeddodecahedron_class::c_cappeddodecahedron	139
c_CappedDodecahedron_doubles3	139
f_cappeddodecahedron_doubles3_class::c_cappeddodecahedron_doubles3	139
c_DividedPolygon	140
f_dividedpolygon_class::c_dividedpolygon	140
f_dodecahedron_class::c_dodecahedron	140
c_Dodecahedron	140
c_ELVIRANeighborhood	141
f_elviraneighborhood_class::c_elviraneighborhood	142
c_ListedVM_VMAN	142
f_listedvm_vman_class::c_listedvm_vman	143
c_LocalizedSeparatorLink	143
f_localizedseparatorlink_class::c_localizedseparatorlink	143
c_LocalizerLink	144
f_localizerlink_class::c_localizerlink	144
c_LVIRANeighborhood_RectangularCuboid	145
f_lviraneighborhood_rectangularcuboid_class::c_lviraneighborhood_rectangularcuboid	145
c_ObjectAllocationServer_LocalizedSeparatorLink	146
f_objectallocationserver_localizedseparatorlink_class::c_objectallocationserver_localizedseparatorlink	146
f_objectallocationserver_localizerlink_class::c_objectallocationserver_localizerlink	146
c_ObjectAllocationServer_LocalizerLink	147
f_objectallocationserver_planarlocalizer_class::c_objectallocationserver_planarlocalizer	147
c_ObjectAllocationServer_PlanarLocalizer	148
c_ObjectAllocationServer_PlanarSeparator	149
f_objectallocationserver_planarseparator_class::c_objectallocationserver_planarseparator	150
f_planarlocalizer_class::c_planarlocalizer	150
c_PlanarLocalizer	150
f_planarseparator_class::c_planarseparator	151
c_PlanarSeparator	151
f_polygon_class::c_polygon	152
c_Polygon	152
f_polyhedron24_class::c_polyhedron24	152
c_Polyhedron24	153
f_polyhedron24_doubles3_class::c_polyhedron24_doubles3	153
c_Polyhedron24_doubles3	153
c_R2PNeighborhood_RectangularCuboid	154
f_r2pneighborhood_rectangularcuboid_class::c_r2pneighborhood_rectangularcuboid	154
c_RectangularCuboid	155
f_rectangularcuboid_class::c_rectangularcuboid	155
c_SepVM	155
f_sepvm_class::c_sepvm	156
c_SepVM_doubles3	156
f_sepvm_doubles3_class::c_sepvm_doubles3	157
c_Tagged_AccumListedVM_VMAN	157
f_tagged_accumlistedvm_vman_class::c_tagged_accumlistedvm_vman	157
f_tagged_accumvm_sepvm_class::c_tagged_accumvm_sepvm	158
c_Tagged_AccumVM_SepVM	158
f_tagged_accumvm_vm_class::c_tagged_accumvm_vm	158
c_Tagged_AccumVM_VM	159
f_tet_class::c_tet	159

c_Tet	159
f_tri_class::c_tri	160
c_Tri	160
f_vman_class::c_vman	160
c_VMAN	161
f_tri_class::calculateandsetplaneofexistence	161
f_polygon_class::calculateandsetplaneofexistence	162
f_dividedpolygon_class::calculateandsetplaneofexistence	162
f_tri_class::calculatecentroid	162
f_polygon_class::calculatecentroid	162
f_polygon_class::calculatenearestptonsurface	163
f_tri_class::calculatenormal	163
f_polygon_class::calculatenormal	163
f_dividedpolygon_class::calculatenormal	163
f_tri_class::calculatesign	164
f_polygon_class::calculatesign	164
f_dividedpolygon_class::calculatesign	164
f_dividedpolygon_class::calculatesurfacearea	164
f_tri_class::calculatevolume	165
f_polygon_class::calculatevolume	165
f_rectangularcuboid_class::calculatevolume	165
f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_type	165
f_cappeddodecahedron_class::cappeddodecahedron_type	166
IRL::CappedDodecahedronSpecialization< Derived, VertexType >	167
IRL::CellCollection< CellGroupType >	
Class that contains a list of cells and coupled to moments	168
IRL::CellGroupedMoments< CellType, ContainedMomentsType >	
A class that couples together a cell and another class	169
IRL::Centroid2D_Functor	171
IRL::Centroid3D_Functor	172
IRL::ChainedBlockStorage< ContainedType, kBlockSize >	173
f_listedvm_vman_class::clear	174
f_tagged_accumlistedvm_vman_class::clear	174
IRL::Collection< ObjectType >	
Essentially a std::vector with assertions on access through operator[] in debug mode	174
IRL::ConcaveBoxSpecialization< Derived, VertexType >	176
IRL::ConstIteratorThroughBracketOperator< ContainerType >	177
f_tet_class::construct	178
f_tri_class::construct	178
f_cappeddodecahedron_class::construct	178
f_dodecahedron_class::construct	179
f_cappeddodecahedron_doubles3_class::construct	179
f_polygon_class::construct	179
f_polyhedron24_class::construct	179
f_polyhedron24_doubles3_class::construct	180
f_dividedpolygon_class::construct	180
f_rectangularcuboid_class::construct	180
f_sepvm_class::construct	180
f_rectangularcuboid_class::construct_2pt	181
f_dividedpolygon_class::constructfrompolygon	181
IRL::ContinueDividingVolumeByPlanarReconstruction< HowToCutSimplexByNextPlane, HowToBranchEarlyIfNoIntersection, HowToBranchEarlyIfIntersection >	
181	
IRL::ContinueReducingVolumeToBeInternalToReconstruction	182
f_planarseparator_class::copy	182
IRL::CutSimplexByNextPlaneAccessedByIndex	182
IRL::CutTetVertices< VertexType >	182
IRL::CutTriangleVerticesAndPlane< VertexType >	183
f_bytbuffer_class::dataptr	184

IRL::DecomposedPolygonVertexStorage< VertexType >	184
IRL::DecomposedPolyhedronVertexStorage< VertexType >	186
IRL::DecomposedPolytopeVertexStorage< VertexType >	187
IRL::DelayedExpandableVertexAccess< Derived, VertexType, kMaskedVertices >	188
f_dividedpolygon_class::dividedpolygon_type	189
IRL::DividedPolygonSpecialization< Derived, VertexType >	
A polygon that knows its centroid and can return the triangles that it is comprised of	189
IRL::DivideSimplexByPlanarReconstruction< HowToHandleVolumeCompletelyInternalToPlanarReconstruction, HowToContinueD	
191	
f_dodecahedron_class::dodecahedron_type	192
IRL::DodecahedronSpecialization< Derived, VertexType >	192
IRL::DoesNotHaveACollection< ReturnType >	193
IRL::DoesNotHaveALocalizer< ReconstructionType >	193
IRL::DoesNotHaveANestedType< ReturnType >	194
IRL::DoesNotHaveAReconstructionLink< ReconstructionType >	194
IRL::DoesNotHaveASeparator< ReconstructionType >	194
IRL::doNoEarlyBranching	195
IRL::earlyBranchBothFullyBelowAndFullyAboveSimplicesLink	195
IRL::earlyBranchBothFullyBelowAndFullyAboveSimplicesNoLink	195
IRL::ELVIRA_2D	196
IRL::ELVIRA_3D	197
IRL::ELVIRADebug< BaseELVIRA >	199
IRL::ELVIRADebug< ELVIRA_2D >	199
IRL::ELVIRADebug< ELVIRA_3D >	200
IRL::ELVIRANeighborhood	
Below plane volume fraction information in a stencil, to be used for ELVIRA. Up to 27 cells to	
cover 3x3x3 stencil in 3D. Can partially fill to 9 for 2D	201
f_elviraneighborhood_class::elviraneighborhood_type	203
f_r2pneighborhood_rectangularcuboid_class::emptyneighborhood	203
f_lviraneighborhood_rectangularcuboid_class::emptyneighborhood	204
f_listedvm_vman_class::erase	204
IRL::ExpandableDividedPolygon< VertexType >	204
IRL::ExpandablePolygon< VertexType >	205
IRL::ExpandablePtList< VertexType >	206
IRL::Expr< E >	208
f_bytbuffer_class::F_BytBuffer_dataPtr	208
f_bytbuffer_class::F_BytBuffer_delete	208
f_bytbuffer_class::F_BytBuffer_getSize	208
f_bytbuffer_class::F_BytBuffer_new	209
f_bytbuffer_class::F_BytBuffer_resetBufferPointer	209
f_bytbuffer_class::F_BytBuffer_setSize	209
f_cappeddodecahedron_class::F_CappedDodecahedron_adjustCapToMatchVolume	209
f_cappeddodecahedron_class::F_CappedDodecahedron_construct	210
f_cappeddodecahedron_class::F_CappedDodecahedron_delete	210
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_adjustCapToMatchVolume	210
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_construct	210
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_delete	211
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getBoundingPts	211
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getData	211
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getPt	211
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_new	212
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_setData	212
f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_setPt	212
f_cappeddodecahedron_class::F_CappedDodecahedron_getBoundingPts	212
f_cappeddodecahedron_class::F_CappedDodecahedron_getPt	213
f_cappeddodecahedron_class::F_CappedDodecahedron_new	213
f_constants::F_Constants_setMinimumSurfaceAreaToTrack	213
f_constants::F_Constants_setMinimumVolumeToTrack	213

f_constants::F_Constants_setVolumeFractionBounds	214
f_constants::F_Constants_setVolumeFractionToleranceForDistanceFinding	214
f_dividedpolygon_class::F_DividedPolygon_calculateAndSetPlaneOfExistence	214
f_dividedpolygon_class::F_DividedPolygon_calculateNormal	214
f_dividedpolygon_class::F_DividedPolygon_calculateSign	215
f_dividedpolygon_class::F_DividedPolygon_calculateSurfaceArea	215
f_dividedpolygon_class::F_DividedPolygon_construct	215
f_dividedpolygon_class::F_DividedPolygon_constructFromPolygon	215
f_dividedpolygon_class::F_DividedPolygon_delete	216
f_dividedpolygon_class::F_DividedPolygon_getBoundingPts	216
f_dividedpolygon_class::F_DividedPolygon_getLocalizer	216
f_dividedpolygon_class::F_DividedPolygon_getNumberOfPts	216
f_dividedpolygon_class::F_DividedPolygon_getNumberOfSimplicesInDecomposition	217
f_dividedpolygon_class::F_DividedPolygon_getPlaneOfExistence	217
f_dividedpolygon_class::F_DividedPolygon_getPt	217
f_dividedpolygon_class::F_DividedPolygon_getSimplexFromDecomposition	217
f_dividedpolygon_class::F_DividedPolygon_new	218
f_dividedpolygon_class::F_DividedPolygon_printToScreen	218
f_dividedpolygon_class::F_DividedPolygon_resetCentroid	218
f_dividedpolygon_class::F_DividedPolygon_reversePtOrdering	218
f_dividedpolygon_class::F_DividedPolygon_setPlaneOfExistence	219
f_dividedpolygon_class::F_DividedPolygon_zeroPolygon	219
f_dodecahedron_class::F_Dodecahedron_construct	219
f_dodecahedron_class::F_Dodecahedron_delete	219
f_dodecahedron_class::F_Dodecahedron_getBoundingPts	220
f_dodecahedron_class::F_Dodecahedron_new	220
f_elviraneighborhood_class::F_ELVIRANeighborhood_delete	220
f_elviraneighborhood_class::F_ELVIRANeighborhood_new	220
f_elviraneighborhood_class::F_ELVIRANeighborhood_setMember	221
f_elviraneighborhood_class::F_ELVIRANeighborhood_setSize	221
f_cutpolygon::F_getPlanePolygonFromReconstruction_RC_DivPoly	221
f_cutpolygon::F_getPlanePolygonFromReconstruction_RC_Poly	221
f_cutpolygon::F_getReconstructionSurfaceArea_RC	222
f_getvolumemoments::F_GNVM_CD_By_LSL_For_SVM	222
f_getvolumemoments::F_GNVM_CD_By_LSL_For_TagAccumVM_SVM	222
f_getvolumemoments::F_GNVM_CDWD3_By_LSL_For_SVMAD3	222
f_getvolumemoments::F_GNVM_D_By_LSL_For_SVM	223
f_getvolumemoments::F_GNVM_D_By_LSL_For_TagAccumVM_SVM	223
f_getvolumemoments::F_GNVM_D_By_PS_For_SVM	223
f_getvolumemoments::F_GNVM_P24_By_LSL_For_SVM	223
f_getvolumemoments::F_GNVM_P24WD3_By_LSL_For_SVMAD3	224
f_getvolumemoments::F_GNVM_Poly_By_PL_For_V	224
f_getvolumemoments::F_GNVM_RC_By_PS_For_SVM	224
f_getvolumemoments::F_GNVM_RC_By_PS_For_V	224
f_getvolumemoments::F_GNVM_Tet_By_LSL_For_SVM	225
f_getvolumemoments::F_GNVM_Tri_By_LL_For_TagAVM_VM	225
f_getvolumemoments::F_GNVM_Tri_By_PL_For_V	225
f_getvolumemoments::F_GVM_CD_By_LSL_For_SVM	225
f_getvolumemoments::F_GVM_D_By_LSL_For_SVM	226
f_getvolumemoments::F_GVM_P24_By_LSL_For_SVM	226
f_getvolumemoments::F_GVM_setMethod	226
f_getvolumemoments::F_GVM_Tri_By_LL_For_TagALVM_VMAN	226
f_geometriccuttinghelpers::F_isPtInternal_PL	227
f_geometriccuttinghelpers::F_isPtInternal_PS	227
f_listedvm_vman_class::F_ListedVM_VMAN_append	227
f_listedvm_vman_class::F_ListedVM_VMAN_clear	227
f_listedvm_vman_class::F_ListedVM_VMAN_delete	228
f_listedvm_vman_class::F_ListedVM_VMAN_erase	228

f_listedvm_vman_class::F_ListedVM_VMAN_getMoments	228
f_listedvm_vman_class::F_ListedVM_VMAN_getSize	228
f_listedvm_vman_class::F_ListedVM_VMAN_new	229
f_listedvm_vman_class::F_ListedVM_VMAN_zeroNormalComponent	229
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_delete	229
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_getId	229
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_new	230
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_newFromObjectAllocationServer	230
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivity	230
f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivityNull	230
f_localizedseparatorlink_class::F_LocalizedSeparatorLink setId	231
f_localizerlink_class::F_LocalizerLink_delete	231
f_localizerlink_class::F_LocalizerLink_getId	231
f_localizerlink_class::F_LocalizerLink_new	231
f_localizerlink_class::F_LocalizerLink_newFromObjectAllocationServer	232
f_localizerlink_class::F_LocalizerLink_setEdgeConnectivity	232
f_localizerlink_class::F_LocalizerLink_setEdgeConnectivityNull	232
f_localizerlink_class::F_LocalizerLink setId	232
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_addMember	233
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_delete	233
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_emptyNeighborhood	233
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_new	233
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_setCenterOfStencil	234
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_setMember	234
f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_setSize . . .	234
f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparatorLink_delete	234
f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparatorLink_new	235
f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_delete	235
f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_new	235
f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_delete	235
f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_new	236
f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_PlanarSeparator_delete	236
f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_PlanarSeparator_new	236
f_planarlocalizer_class::F_PlanarLocalizer_addPlane	236
f_planarlocalizer_class::F_PlanarLocalizer_delete	237
f_planarlocalizer_class::F_PlanarLocalizer_new	237
f_planarlocalizer_class::F_PlanarLocalizer_newFromObjectAllocationServer	237
f_planarlocalizer_class::F_PlanarLocalizer_printToScreen	237
f_planarlocalizer_class::F_PlanarLocalizer_setFromRectangularCuboid	238
f_planarlocalizer_class::F_PlanarLocalizer_setNumberOfPlanes	238
f_planarlocalizer_class::F_PlanarLocalizer_setPlane	238
f_planarseparator_class::F_PlanarSeparator_addPlane	238
f_planarseparator_class::F_PlanarSeparator_copy	239
f_planarseparator_class::F_PlanarSeparator_delete	239
f_planarseparator_class::F_PlanarSeparator_getNumberOfPlanes	239
f_planarseparator_class::F_PlanarSeparator_getPlane	239
f_planarseparator_class::F_PlanarSeparator_isFlipped	240
f_planarseparator_class::F_PlanarSeparator_new	240
f_planarseparator_class::F_PlanarSeparator_newFromObjectAllocationServer	240
f_planarseparator_class::F_PlanarSeparator_printToScreen	240
f_planarseparator_class::F_PlanarSeparator_setNumberOfPlanes	241
f_planarseparator_class::F_PlanarSeparator_setPlane	241
f_polygon_class::F_Polygon_calculateAndSetPlaneOfExistence	241
f_polygon_class::F_Polygon_calculateCentroid	241

f_polygon_class::F_Polygon_calculateNearestPtOnSurface	242
f_polygon_class::F_Polygon_calculateNormal	242
f_polygon_class::F_Polygon_calculateSign	242
f_polygon_class::F_Polygon_calculateVolume	242
f_polygon_class::F_Polygon_construct	243
f_polygon_class::F_Polygon_delete	243
f_polygon_class::F_Polygon_getBoundingPts	243
f_polygon_class::F_Polygon_getLocalizer	243
f_polygon_class::F_Polygon_getNumberOfPts	244
f_polygon_class::F_Polygon_getNumberOfSimplicesInDecomposition	244
f_polygon_class::F_Polygon_getPlaneOfExistence	244
f_polygon_class::F_Polygon_getPt	244
f_polygon_class::F_Polygon_getSimplexFromDecomposition	245
f_polygon_class::F_Polygon_new	245
f_polygon_class::F_Polygon_printToScreen	245
f_polygon_class::F_Polygon_reversePtOrdering	245
f_polygon_class::F_Polygon_setPlaneOfExistence	246
f_polygon_class::F_Polygon_zeroPolygon	246
f_polyhedron24_class::F_Polyhedron24_adjustCapToMatchVolume	246
f_polyhedron24_class::F_Polyhedron24_construct	246
f_polyhedron24_class::F_Polyhedron24_delete	247
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_adjustCapToMatchVolume	247
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_construct	247
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_delete	247
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getBoundingPts	248
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getData	248
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getPt	248
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_new	248
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setData	249
f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setPt	249
f_polyhedron24_class::F_Polyhedron24_getBoundingPts	249
f_polyhedron24_class::F_Polyhedron24_getPt	249
f_polyhedron24_class::F_Polyhedron24_new	250
f_polyhedron24_class::F_Polyhedron24_setPt	250
f_r2neighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_addMember	250
f_r2neighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_delete	250
f_r2neighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_emptyNeighborhood	251
f_r2neighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_new	251
f_r2neighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setCenterOfStencil	251
f_r2neighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setMember	251
f_r2neighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setSize	252
f_r2neighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setSurfaceArea	252
f_reconstructioninterface::F_reconstructionWithAdvectionNormals_ListedVM_VMAN_RC	252
f_reconstructioninterface::F_reconstructionWithAdvectionNormalsDebug_ListedVM_VMAN_RC	252
f_reconstructioninterface::F_reconstructionWithELVIRA2D	253
f_reconstructioninterface::F_reconstructionWithELVIRA3D	253
f_reconstructioninterface::F_reconstructionWithLVIRA2D_RC	253
f_reconstructioninterface::F_reconstructionWithLVIRA3D_RC	253
f_reconstructioninterface::F_reconstructionWithMOF2D_RectangularCuboid	254
f_reconstructioninterface::F_reconstructionWithMOF2D_Tri	254
f_reconstructioninterface::F_reconstructionWithMOF2DGiveWeights_RectangularCuboid	254
f_reconstructioninterface::F_reconstructionWithMOF2DGiveWeights_Tri	254
f_reconstructioninterface::F_reconstructionWithMOF3D_RectangularCuboid	255
f_reconstructioninterface::F_reconstructionWithMOF3D_Tet	255
f_reconstructioninterface::F_reconstructionWithMOF3DGiveWeights_RectangularCuboid	255
f_reconstructioninterface::F_reconstructionWithMOF3DGiveWeights_Tet	255

f_reconstructioninterface::F_reconstructionWithR2P2D_RC	256
f_reconstructioninterface::F_reconstructionWithR2P2DDebug_RC	256
f_reconstructioninterface::F_reconstructionWithR2P3D_RC	256
f_reconstructioninterface::F_reconstructionWithR2P3DDebug_RC	256
f_rectangularcuboid_class::F_RectangularCuboid_calculateVolume	257
f_rectangularcuboid_class::F_RectangularCuboid_construct	257
f_rectangularcuboid_class::F_RectangularCuboid_construct_2pt	257
f_rectangularcuboid_class::F_RectangularCuboid_delete	257
f_rectangularcuboid_class::F_RectangularCuboid_getBoundingPts	258
f_rectangularcuboid_class::F_RectangularCuboid_new	258
f_sepvm_class::F_SepVM_construct	258
f_sepvm_class::F_SepVM_delete	258
f_sepvm_doubles3_class::F_SepVM_doubles3_delete	259
f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroid	259
f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroidPtr	259
f_sepvm_doubles3_class::F_SepVM_doubles3_getData	259
f_sepvm_doubles3_class::F_SepVM_doubles3_getVolume	260
f_sepvm_doubles3_class::F_SepVM_doubles3_getVolumePtr	260
f_sepvm_doubles3_class::F_SepVM_doubles3_multiplyByVolume	260
f_sepvm_doubles3_class::F_SepVM_doubles3_new	260
f_sepvm_doubles3_class::F_SepVM_doubles3_normalizeByVolume	261
f_sepvm_class::F_SepVM_getCentroid	261
f_sepvm_class::F_SepVM_getCentroidPtr	261
f_sepvm_class::F_SepVM_getVolume	261
f_sepvm_class::F_SepVM_getVolumePtr	262
f_sepvm_class::F_SepVM_multiplyByVolume	262
f_sepvm_class::F_SepVM_new	262
f_sepvm_class::F_SepVM_normalizeByVolume	262
f_serializer::F_Serializer_serializeAndPack_PlanarSeparator_ByteBuffer	263
f_serializer::F_Serializer_unpackAndStore_PlanarSeparator_ByteBuffer	263
f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS	263
f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS_DefTol	263
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_append	264
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_clear	264
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_delete	264
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getListAtIndex	264
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getSize	265
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getTagForIndex	265
f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_new	265
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_delete	265
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtIndex	266
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtTag	266
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidPtrAtIndex	266
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getSize	266
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getTagForIndex	267
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtIndex	267
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtTag	267
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumePtrAtIndex	267
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_multiplyByVolume	268
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_new	268
f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_normalizeByVolume	268
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_delete	268
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidAtIndex	269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidPtrAtIndex	269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getSize	269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getTagForIndex	269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumeAtIndex	270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumePtrAtIndex	270

f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_multiplyByVolume	270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_new	270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_normalizeByVolume	271
f_tet_class::F_Tet_construct	271
f_tet_class::F_Tet_delete	271
f_tet_class::F_Tet_getBoundingPts	271
f_tet_class::F_Tet_new	272
f_tri_class::F_Tri_calculateAndSetPlaneOfExistence	272
f_tri_class::F_Tri_calculateCentroid	272
f_tri_class::F_Tri_calculateNormal	272
f_tri_class::F_Tri_calculateSign	273
f_tri_class::F_Tri_calculateVolume	273
f_tri_class::F_Tri_construct	273
f_tri_class::F_Tri_delete	273
f_tri_class::F_Tri_getBoundingPts	274
f_tri_class::F_Tri_getLocalizer	274
f_tri_class::F_Tri_getPlaneOfExistence	274
f_tri_class::F_Tri_getVertices	274
f_tri_class::F_Tri_new	275
f_tri_class::F_Tri_reversePtOrdering	275
f_tri_class::F_Tri_setPlaneOfExistence	275
f_vman_class::F_VMAN_delete	275
f_vman_class::F_VMAN_getCentroid	276
f_vman_class::F_VMAN_getNormal	276
f_vman_class::F_VMAN_getVolume	276
f_vman_class::F_VMAN_multiplyByVolume	276
f_vman_class::F_VMAN_new	277
f_vman_class::F_VMAN_normalizeByVolume	277
IRL::Face< HalfEdgeType >	277
IRL::GeneralPolygon< Derived, VertexType, SimplexType >	279
IRL::GeneralPolyhedron< Derived, VertexType, SimplexType >	281
f_cappeddodecahedron_class::getboundingpts	283
f_tet_class::getboundingpts	283
f_tri_class::getboundingpts	284
f_dodecahedron_class::getboundingpts	284
f_cappeddodecahedron_doubles3_class::getboundingpts	284
f_polygon_class::getboundingpts	284
f_polyhedron24_class::getboundingpts	285
f_polyhedron24_doubles3_class::getboundingpts	285
f_rectangularcuboid_class::getboundingpts	285
f_dividedpolygon_class::getboundingpts	285
f_vman_class::getcentroid	286
f_sepvm_class::getcentroid	286
f_sepvm_doubles3_class::getcentroid	286
f_tagged_accumvm_sepvm_class::getcentroidatindex	286
f_tagged_accumvm_vm_class::getcentroidatindex	287
f_tagged_accumvm_sepvm_class::getcentroidattag	287
f_sepvm_class::getcentroidptr	287
f_sepvm_doubles3_class::getcentroidptr	287
f_tagged_accumvm_vm_class::getcentroidptratindex	288
f_tagged_accumvm_sepvm_class::getcentroidptratindex	288
f_objectallocationserver_localizedseparatorlink_class::getobject	288
f_tagged_accumvm_sepvm_class::getobject	288
f_objectallocationserver_localizerlink_class::getobject	289
f_objectallocationserver_planarlocalizer_class::getobject	289
f_tagged_accumvm_vm_class::getobject	289
f_objectallocationserver_planarseparator_class::getobject	289
f_planarlocalizer_class::getobject	290

f_tet_class::getcobject	290
f_tri_class::getcobject	290
f_planarseparator_class::getcobject	290
f_dodecahedron_class::getcobject	291
f_vman_class::getcobject	291
f_cappeddodecahedron_doubles3_class::getcobject	291
f_elviraneighborhood_class::getcobject	291
f_polygon_class::getcobject	292
f_polyhedron24_class::getcobject	292
f_polyhedron24_doubles3_class::getcobject	292
f_listedvm_vman_class::getcobject	292
f_bytbuffer_class::getcobject	293
f_r2pnighborhood_rectangularcuboid_class::getcobject	293
f_localizedseparatorlink_class::getcobject	293
f_dividedpolygon_class::getcobject	293
f_rectangularcuboid_class::getcobject	294
f_localizerlink_class::getcobject	294
f_sepvm_class::getcobject	294
f_sepvm_doubles3_class::getcobject	294
f_lviraneighborhood_rectangularcuboid_class::getcobject	295
f_tagged_accumlistedvm_vman_class::getcobject	295
f_cappeddodecahedron_class::getcobject	295
f_cappeddodecahedron_doubles3_class::getdata	295
f_polyhedron24_doubles3_class::getdata	296
f_sepvm_doubles3_class::getdata	296
f_localizedseparatorlink_class::getid	296
f_localizerlink_class::getid	296
f_tagged_accumlistedvm_vman_class::getlistatindex	297
f_tri_class::getlocalizer	297
f_polygon_class::getlocalizer	297
f_dividedpolygon_class::getlocalizer	297
f_listedvm_vman_class::getmoments	298
f_vman_class::getnormal	298
f_getvolumemoments::getnormalizedvolumemoments	298
f_planarseparator_class::getnumberofplanes	299
f_polygon_class::getnumberofsimplicesindecomposition	299
f_dividedpolygon_class::getnumberofsimplicesindecomposition	299
f_polygon_class::getnumberofvertices	299
f_dividedpolygon_class::getnumberofvertices	300
f_planarseparator_class::getplane	300
f_tri_class::getplaneofexistence	300
f_polygon_class::getplaneofexistence	300
f_dividedpolygon_class::getplaneofexistence	301
f_cutpolygon::getplanepolygonfromreconstruction	301
f_cappeddodecahedron_class::getpt	301
f_cappeddodecahedron_doubles3_class::getpt	301
f_polygon_class::getpt	302
f_polyhedron24_class::getpt	302
f_polyhedron24_doubles3_class::getpt	302
f_dividedpolygon_class::getpt	302
f_cutpolygon::getreconstructionsurfacearea	303
f_polygon_class::getsimplexfromdecomposition	303
f_dividedpolygon_class::getsimplexfromdecomposition	303
f_tagged_accumvm_vm_class::getsize	303
f_tagged_accumvm_sepvm_class::getsize	304
f_listedvm_vman_class::getsize	304
f_bytbuffer_class::getsize	304
f_tagged_accumlistedvm_vman_class::getsize	304

f_tagged_accumvm_sepvm_class::gettagforindex	305
f_tagged_accumvm_vm_class::gettagforindex	305
f_tagged_accumlistedvm_vman_class::gettagforindex	305
f_tri_class::getvertices	305
f_vman_class::getvolume	306
f_sepvm_class::getvolume	306
f_sepvm_doubles3_class::getvolume	306
f_tagged_accumvm_sepvm_class::getvolumeatindex	306
f_tagged_accumvm_vm_class::getvolumeatindex	307
f_tagged_accumvm_sepvm_class::getvolumeattag	307
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, Enable	
Given a polyhedron and a localizer or separator return the un-normalized ReturnType. (Return←	
Type can be volume, VolumeMoments, or SeparatedMoments<VolumeMoments>)	307
f_getvolumemoments::getvolumemoments	308
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, NullReconstruction, enable	
308	
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, PlanarSeparator, enable_if	
309	
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable	
310	
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable	
310	
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable	
311	
f_getvolumemoments::getvolumemoments_setmethod	311
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfE	
311	
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfE	
312	
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfE	
312	
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfE	
314	
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfE	
315	
IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfE	
315	
f_sepvm_class::getvolumeptr	316
f_sepvm_doubles3_class::getvolumeptr	316
f_tagged_accumvm_vm_class::getvolumepratindex	316
f_tagged_accumvm_sepvm_class::getvolumepratindex	316
IRL::HalfEdge< VertexType >	317
IRL::HalfEdgeCutting	318
IRL::HalfEdgePolygon< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMax	
318	
IRL::HalfEdgePolyhedron< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMax	
320	
IRL::HalfEdgePolytope< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMax	
321	
IRL::has_localizer< C >	322
IRL::has_localizer< const C >	323
IRL::has_localizer< LocalizedSeparator >	324
IRL::has_localizer< LocalizedSeparatorGroup >	325
IRL::has_localizer< LocalizedSeparatorGroupLink >	326
IRL::has_localizer< LocalizedSeparatorLink >	327
IRL::has_localizer< LocalizerLink >	328
IRL::has_localizer< LocalizerLinkFromLocalizedSeparatorLink >	329
IRL::has_localizer< MaskedLocalizedSeparatorLink >	330

IRL::has_localizer< PlanarLocalizer >	331
IRL::has_localizer< RestrictedLocalizerLinkFromLocalizedSeparatorLink >	332
IRL::has_localizer< RestrictedMaskedLocalizedSeparatorLink >	333
IRL::has_null_reconstruction< C >	334
IRL::has_null_reconstruction< const C >	335
IRL::has_null_reconstruction< NullReconstruction >	336
IRL::has_separator< C >	337
IRL::has_separator< const C >	338
IRL::has_separator< LocalizedSeparator >	339
IRL::has_separator< LocalizedSeparatorGroup >	340
IRL::has_separator< LocalizedSeparatorGroupLink >	341
IRL::has_separator< LocalizedSeparatorLink >	342
IRL::has_separator< MaskedLocalizedSeparatorLink >	343
IRL::has_separator< PlanarSeparator >	344
IRL::has_separator< PlanarSeparatorPath >	345
IRL::has_separator< RestrictedMaskedLocalizedSeparatorLink >	346
IRL::HasACollection< ReturnType >	347
IRL::HasALocalizer< ReconstructionType >	347
IRL::HasANestedType< ReturnType >	348
IRL::HasAReconstructionLink< ReconstructionType >	348
IRL::HasASeparator< ReconstructionType >	348
IRL::HasLocalizer_AND_HasSeparator< ReconstructionType >	349
IRL::HasLocalizer_OR_HasSeparator< ReconstructionType >	349
IRL::HasLocalizer_XOR_HasSeparator< ReconstructionType >	350
IRL::HexahedronSpecialization< Derived, VertexType >	
A hexahedron class	350
IRL::IgnoreVolumeAbovePlane	351
IRL::Illinois< OptimizingClass >	
A templated class driver that performs the Illinois method of Regula-Falsi zero-finding	351
IRL::is_moments_collection< C >	
Type trait to allow static checking that an object is a AccumulatedVolumeMoments	353
IRL::is_moments_collection< AccumulatedListedVolumeMoments< VolumeMomentsType > >	
Any instantiation of AccumulatedListedVolumeMoments is a type of volumeMomentsList	355
IRL::is_moments_collection< AccumulatedVolumeMoments< VolumeMomentsType > >	
Any instantiation of AccumulatedVolumeMoments is a type of volumeMomentsList	356
IRL::is_moments_collection< const C >	357
IRL::is_moments_collection< ListedVolumeMoments< VolumeMomentsType > >	
Any instantiation of ListedVolumeMoments is a type of volumeMomentsList	358
IRL::is_moments_collection< TaggedAccumulatedListedVolumeMoments< VolumeMomentsType > >	
Any instantiation of AccumulatedListedVolumeMoments is a type of volumeMomentsList	359
IRL::is_moments_collection< TaggedAccumulatedVolumeMoments< VolumeMomentsType > >	360
IRL::is_nested_moments< C >	
Type trait to allow static checking that an object is a AccumulatedVolumeMoments	360
IRL::is_nested_moments< AccumulateWrapper< NestedType > >	
Any instantiation of AccumulatedVolumeMoments is a type of volumeMomentsList	362
IRL::is_nested_moments< const C >	363
IRL::is_planar_separator< C >	364
IRL::is_planar_separator< const C >	365
IRL::is_planar_separator< PlanarSeparator >	366
IRL::is_polygon< C >	367
IRL::is_polygon< const C >	368
IRL::is_polygon< ExpandableDividedPolygon< VertexType > >	369
IRL::is_polygon< ExpandablePolygon< VertexType > >	370
IRL::is_polygon< ProxyTri< GeometryType > >	371
IRL::is_polygon< SegmentedDecomposedPolygon< VertexType > >	372
IRL::is_polygon< SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices > >	373
IRL::is_polygon< StoredTri< VertexType > >	374
IRL::is_polyhedron< C >	375

IRL::is_polyhedron< const C >	376
IRL::is_polyhedron< ProxyTet< GeometryType > >	377
IRL::is_polyhedron< SegmentedDecomposedPolyhedron< VertexType > >	378
IRL::is_polyhedron< SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices > >	
379	
IRL::is_polyhedron< StoredCappedDodecahedron< VertexType > >	380
IRL::is_polyhedron< StoredConcaveBox< VertexType > >	381
IRL::is_polyhedron< StoredDodecahedron< VertexType > >	382
IRL::is_polyhedron< StoredHexahedron< VertexType > >	383
IRL::is_polyhedron< StoredPolyhedron24< VertexType > >	384
IRL::is_polyhedron< StoredRectangularCuboid< VertexType > >	385
IRL::is_polyhedron< StoredTet< VertexType > >	386
IRL::is_reconstruction_link< C >	387
IRL::is_reconstruction_link< const C >	388
IRL::is_reconstruction_link< LocalizedSeparatorGroupLink >	389
IRL::is_reconstruction_link< LocalizedSeparatorLink >	390
IRL::is_reconstruction_link< LocalizerLink >	391
IRL::is_reconstruction_link< LocalizerLinkFromLocalizedSeparatorLink >	392
IRL::is_reconstruction_link< MaskedLocalizedSeparatorLink >	393
IRL::is_reconstruction_link< PlanarSeparatorPath >	394
IRL::is_reconstruction_link< RestrictedLocalizerLinkFromLocalizedSeparatorLink >	395
IRL::is_reconstruction_link< RestrictedMaskedLocalizedSeparatorLink >	396
IRL::is_separated_moments< C >	
Marking of if volume	396
IRL::is_separated_moments< const C >	398
IRL::is_separated_moments< SeparatedMoments< MomentsType > >	399
IRL::is_tet< C >	400
IRL::is_tet< const C >	401
IRL::is_tet< ProxyTet< GeometryType > >	402
IRL::is_tet< StoredTet< VertexType > >	403
IRL::is_tri< C >	404
IRL::is_tri< const C >	405
IRL::is_tri< ProxyTri< GeometryType > >	406
IRL::is_tri< StoredTri< VertexType > >	407
f_planarseparator_class::isflipped	407
IRL::isHalfEdgeCutting< C >	408
IRL::isHalfEdgeCutting< const C >	409
IRL::isHalfEdgeCutting< HalfEdgeCutting >	410
IRL::IsNotNullReconstruction< ReconstructionType >	410
IRL::IsNotAPlanarSeparator< ReconstructionType >	411
IRL::IsNullReconstruction< ReconstructionType >	411
IRL::IsPlanarSeparator< ReconstructionType >	412
f_geometriccuttinghelpers::isptinternal	412
IRL::isRecursiveSimplexCutting< C >	412
IRL::isRecursiveSimplexCutting< const C >	413
IRL::isRecursiveSimplexCutting< RecursiveSimplexCutting >	414
IRL::isSimplexCutting< C >	415
IRL::isSimplexCutting< const C >	416
IRL::isSimplexCutting< SimplexCutting >	417
IRL::IterativeSolverForDistance< CellType, kMaxPlanes >	
Volume conserving distance-finding routine for two-plane reconstructions wrapped in a class	418
IRL::IteratorThroughBracketOperator< ContainerType >	422
IRL::JoinedReconstructions< CurrentReconstruction, NextReconstruction >	
Class that ties together two reconstructions. Will first cut by and use the current reconstruction, and then pass onwards to the next reconstruction	423
IRL::JoinedReconstructions< CurrentReconstruction, NullReconstruction >	424
IRL::JoinedReconstructionsToGroup< CurrentReconstruction, GroupReconstruction >	425

IRL::KMeans	A class that executes Kmeans when provided an appropriate DrivingClass	425
IRL::LevenbergMarquardt< OptimizingClass, kRows, kColumns >	Levenberg-Marquardt optimization routine	426
IRL::LevenbergMarquardt< OptimizingClass, -1, kColumns >	429
IRL::LinearInterpolation_Functor	430
f_listedvm_vman_class::listedvm_vman_type	430
IRL::ListedVolumeMoments< VolumeMomentsType >	VolumeMoments wrapper for Lister class to add ability to normalize	431
IRL::Lister< ObjectType >	This class is a Collection that has operator += add the other Lister or object to the back of its own collection	433
f_localizedseparatorlink_class::localizedseparatorlink_type	434
f_localizerlink_class::localizerlink_type	435
IRL::LocalizerLinkFromLocalizedSeparatorLink	435
IRL::LVIRA_2D< CellType >	LVIRA class for reconstructions in 2D (x-y plane)	436
IRL::LVIRA_3D< CellType >	LVIRA class for reconstructions in 3 dimensions	439
IRL::LVIRACommon< CellType, kColumns >	Class to contain data and methods that will be used in all of the specific LVIRA optimization classes	442
IRL::LVIRADebug< LVIRAType >	This class just calls the LVIRAType functions but allows debug statements to be printed. The solution path is also saved to be exported and visualized	446
IRL::LVIRANeighborhood< CellType >	Neighborhood storage used in the LVIRA optimization routines. This stores the CellGroupedMoments of the cell and the volume fraction	447
f_lviranighborhood_rectangularcuboid_class::lviranighborhood_rectangularcuboid_type	450
IRL::MaskedLocalizedSeparatorLink	450
IRL::MaskStripper< MaskedType >	451
IRL::MathVector< kNumberOfElements >	452
IRL::MOF_2D< CellType >	453
IRL::MOF_3D< CellType >	455
IRL::MOFCommon< CellType >	456
IRL::MOFDebug< MOFType >	This class just calls the MOFType functions but allows debug statements to be printed	459
IRL::MomentCalculationType< VertexType, ProxyType, Enable >	461
IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polygon< ProxyType >::value > >	461	
IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polyhedron< ProxyType >::value > >	461	
IRL::MultiMaterialAccumulatedVolumeMoments< MomentsType >	This class is a specialization of TaggedAccumulatedVolumeMoments< MomentsType > that allows the computation of SeparatedMoments< MomentsType > for the target ID and the encompassing volume. Encompassing volume can be either given, or computed as the moments of all other phases	462
f_tagged_accumvm_sepvm_class::multiplybyvolume	463
f_tagged_accumvm_vm_class::multiplybyvolume	464
f_vman_class::multiplybyvolume	464
f_sepvm_class::multiplybyvolume	464
f_sepvm_doubles3_class::multiplybyvolume	464
f_objectallocationserver_localizedseparatorlink_class::new	465
f_elviraneighborhood_class::new	465
f_objectallocationserver_planarlocalizer_class::new	465
f_planarlocalizer_class::new	465
f_tet_class::new	466
f_polygon_class::new	466

f_dodecahedron_class::new	466
f_planarseparator_class::new	466
f_rectangularcuboid_class::new	467
f_vman_class::new	467
f_tri_class::new	467
f_cappeddodecahedron_doubles3_class::new	467
f_tagged_accumvm_vm_class::new	468
f_tagged_accumlistedvm_vman_class::new	468
f_dividedpolygon_class::new	468
f_polyhedron24_class::new	468
f_polyhedron24_doubles3_class::new	469
f_listedvm_vman_class::new	469
f_bytbuffer_class::new	469
f_objectallocationserver_planarseparator_class::new	469
f_r2neighborhood_rectangularcuboid_class::new	470
f_localizedseparatorlink_class::new	470
f_localizerlink_class::new	470
f_objectallocationserver_localizerlink_class::new	470
f_sepvm_class::new	471
f_sepvm_doubles3_class::new	471
f_lviraneighborhood_rectangularcuboid_class::new	471
f_tagged_accumvm_sepvm_class::new	471
f_cappeddodecahedron_class::new	472
IRL::Normal	
A normal vector in 3D space	472
f_tagged_accumvm_sepvm_class::normalizebyvolume	474
f_tagged_accumvm_vm_class::normalizebyvolume	474
f_sepvm_class::normalizebyvolume	475
f_sepvm_doubles3_class::normalizebyvolume	475
f_vman_class::normalizebyvolume	475
IRL::NullReconstruction	
IRL::ObjectAllocationServer< ObjectType >	476
f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_type	476
f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_type	477
f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer_type	477
f_objectallocationserver_planarseparator_class::objectallocationserver_planarseparator_type	478
IRL::PartitionByNormal< MomentsContainerType >	
This is a class that takes a list of objects and separates the list into two partitions, which are represented by the sums of the objects	478
IRL::PassToNestedType	
IRL::PathGraphNode< NodeType >	
Class to inherit from to provide linking between reconstructions, where the linking exists P ← ER RECONSTRUCTION. This also follows a directed path-graph (linear graph), with a starting node and constant direction to the end node. The interface is kept the same as for the UnDirectedGraphNode in order to facilitate reuse in the cutting routines. All neighbor indices will simply point to the next node in the path-graph	480
IRL::PlanarLocalizer	
A planar representation of a convex polyhedron to localize integrations	482
f_planarlocalizer_class::planarlocalizer_type	484
IRL::PlanarReconstruction< kStackPlanes >	
IRL::PlanarSeparator	
f_planarseparator_class::planarseparator_type	489
IRL::PlanarSeparatorPathGroup	
This is a class that helps organize PlanarSeparatorPath objects to be coherent in memory and facilitates their control more easily. It will continue to store all PlanarSeparatorPath objects that were added to it, but in most routine only the active ones will be used. Here, active means those that are present in priority_order_by_id_m	490

IRL::Plane	
Plane defined by $\mathbf{n} \cdot \mathbf{x} - d = 0$, where the normal points from liquid to gas	491
f_polygon_class::polygon_type	492
IRL::PolygonBase< VertexType >	493
IRL::PolygonConstructors< Derived, VertexType, StorageType, PlaneOfExistenceType >	493
IRL::PolygonMomentsCalculation< Derived, VertexType, SimplexType >	494
IRL::PolygonMomentsCalculationCommon< Derived, VertexType, SimplexType >	495
IRL::PolygonSpecialization< Derived, VertexType >	
A polygon from a given plane	496
IRL::Polyhedron< VertexType, kMaxNumberOfPts >	499
f_polyhedron24_doubles3_class::polyhedron24_doubles3_type	499
f_polyhedron24_class::polyhedron24_type	500
IRL::Polyhedron24Specialization< Derived, VertexType >	500
IRL::PolyhedronMomentsCalculation< Derived, VertexType, SimplexType >	501
IRL::PolyhedronMomentsCalculation< Derived, PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, SimplexType >	502
IRL::PolyhedronMomentsCalculationCommon< Derived, VertexType, SimplexType >	503
f_dividedpolygon_class::printtoscreen	504
f_planarlocalizer_class::printtoscreen	504
f_planarseparator_class::printtoscreen	504
f_polygon_class::printtoscreen	505
IRL::ProxyPolygonConstructors< Derived, GeometryType, kNumberOfVertices, PlaneOfExistenceType >	505
IRL::ProxyTet< GeometryType >	
A tet from a collection of 4 points	506
IRL::ProxyTri< GeometryType >	507
IRL::ProxyVertexAccess< Derived, GeometryType, kNumberOfVertices >	508
IRL::Pt	
A point in 3D space	509
IRL::PtList< VertexType, kMaxNumberOfPts >	513
IRL::PtWithDataCommon< Derived, AttachedDataType >	514
IRL::PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >	515
IRL::R2P_2D1P< CellType >	
R2P class for reconstructions in 2 dimensions with 1 plane (hence 2D1P)	516
IRL::R2P_2D2P< CellType >	
R2P class for reconstructions in 2 dimensions with 2 plane (hence 2D2P)	519
IRL::R2P_3D1P< CellType >	
R2P class for reconstructions in 3 dimensions with 1 plane (hence 3D1P)	521
IRL::R2P_3D2P< CellType >	
R2P class for reconstructions in 3 dimensions with 2 plane (hence 3D2P)	524
IRL::R2PCommon< CellType, kColumns >	
Class to contain data and methods that will be used in all of the specific R2Poptimization	527
IRL::R2PDebug< R2PType >	
This class just calls the R2PType functions but allows debug statements to be printed	532
IRL::R2PNeighborhood< CellType >	
Neighborhood storage used in the R2P optimization routines	534
f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_type	536
short_alloc::short_alloc< T, N, Align >::rebind< _Up >	537
IRL::ReconstructionLink< ReconstructionType, GraphType >	
This is a class template for linking together reconstructions that are already pointers to something else. This then does not require a different constructor because we are already taking pointers of the underlying reconstructions. For classes that are currently storing their own data, it would probably be best to generate a wrapping pointer class, which can be done with Joined<→ Reconstruction<ClassYouWant, NullReconstruction>	537
f_reconstructioninterface::reconstructionwithadvectednormals	538
f_reconstructioninterface::reconstructionwithadvectednormalsdebug	539
f_reconstructioninterface::reconstructionwithlvira2d	539
f_reconstructioninterface::reconstructionwithlvira3d	539

f_reconstructioninterface::reconstructionwithmof2d	539
f_reconstructioninterface::reconstructionwithmof3d	540
f_reconstructioninterface::reconstructionwithr2p2d	540
f_reconstructioninterface::reconstructionwithr2p2ddebug	540
f_reconstructioninterface::reconstructionwithr2p3d	540
f_reconstructioninterface::reconstructionwithr2p3ddebug	541
f_rectangularcuboid_class::rectangularcuboid_type	541
IRL::RectangularCuboidBase< VertexType >	541
IRL::RectangularCuboidCommon< Derived, VertexType >	542
IRL::RectangularCuboidSpecialization< Derived, VertexType >	
A rectangular cuboid	542
IRL::RecursiveSimplexCutting	543
IRL::ReferenceFrame	
A reference frame with three normals (1 for each direction in 3D space)	543
IRL::ReferredToPlaneOFExistence	544
f_bytbuffer_class::resetbufferpointer	544
f_dividedpolygon_class::resetcentroid	545
IRL::RestrictedLocalizerLinkFromLocalizedSeparatorLink	545
IRL::RestrictedMaskedLocalizedSeparatorLink	546
f_tri_class::reverseptordering	547
f_dividedpolygon_class::reverseptordering	547
f_polygon_class::reverseptordering	547
IRL::Secant< OptimizingClass >	
A templated class driver that performs the Secant method for optimization	548
IRL::SegmentedDecomposedPolygon< VertexType >	551
IRL::SegmentedDecomposedPolyhedron< VertexType >	552
IRL::SegmentedDecomposedPolytope< VertexStorageType, VertexType, ProxyType >	553
IRL::SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices >	554
IRL::SegmentedHalfEdgePolygonCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >	555
IRL::SegmentedHalfEdgePolygonSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices >	
556	
IRL::SegmentedHalfEdgePolygonSpecificPt< Pt, FaceType, VertexType, kMaxFaces, kMaxVertices >	556
IRL::SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices >	556
IRL::SegmentedHalfEdgePolyhedronCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >	557
IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices >	
558	
IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, FaceType, Ver	
559	
IRL::SegmentedHalfEdgePolytope< FaceType, VertexType, kMaxFaces, kMaxVertices >	559
IRL::SelfExpandingCollection< ObjectType >	
A modified Collection class that self expands when using the operator[] would force out of mem-	
ory access	561
IRL::SeparatedMoments< MomentsType >	562
IRL::SeparatedMoments< VolumeMoments >	562
IRL::SeparatedMoments< VolumeMomentsAndDoubles< kArrayLength > >	563
IRL::SeparatedMomentsCommon< Derived, MomentsType >	
Storage for multiple volume moments	564
f_sepvm_doubles3_class::sepvm_doubles3_type	568
f_sepvm_class::sepvm_type	568
f_serializer::serializeandpack	569
f_lviraneighborhood_rectangularcuboid_class::setcenterofstencil	569
f_r2pneighborhood_rectangularcuboid_class::setcenterofstencil	569
f_polyhedron24_doubles3_class::setdata	570
f_cappeddodecahedron_doubles3_class::setdata	570
f_volumefractionmatching::setdistancetomatchvolumefraction	570
f_localizedseparatorlink_class::setedgeconnectivity	570
f_localizerlink_class::setedgeconnectivity	571
f_localizedseparatorlink_class::setedgeconnectivitynull	571

f_localizerlink_class::setedgeconnectivitynull	571
f_planarlocalizer_class::setfromrectangularcuboid	571
f_localizerlink_class::setid	572
f_localizedseparatorlink_class::setid	572
f_elviraneighborhood_class::setmember	572
f_r2pneighborhood_rectangularcuboid_class::setmember	572
f_lviraneighborhood_rectangularcuboid_class::setmember	573
f_planarseparator_class::setnumberofplanes	573
f_planarlocalizer_class::setnumberofplanes	573
f_planarlocalizer_class::setplane	573
f_planarseparator_class::setplane	574
f_tri_class::setplaneofexistence	574
f_polygon_class::setplaneofexistence	574
f_dividedpolygon_class::setplaneofexistence	574
f_polyhedron24_doubles3_class::setpt	575
f_cappeddodecahedron_doubles3_class::setpt	575
f_polyhedron24_class::setpt	575
f_elviraneighborhood_class::setsize	575
f_lviraneighborhood_rectangularcuboid_class::setsize	576
f_r2pneighborhood_rectangularcuboid_class::setsize	576
f_bytebuffer_class::setsize	576
f_r2pneighborhood_rectangularcuboid_class::setsurfacearea	576
IRL::ShareVolumeAbovePlaneWithLinkedNeighbor	577
short_alloc::short_alloc< T, N, Align >	577
IRL::SimplexCutting	578
IRL::SimplexWrapper< SimplexType, Enable >	578
IRL::SimplexWrapper< SimplexType, enable_if_t< is_tet< SimplexType >::value > >	578
IRL::SimplexWrapper< SimplexType, enable_if_t< is_tri< SimplexType >::value > >	580
IRL::SmallVector< T, BuffSizelnElements >	580
IRL::StackVector< ObjectType, kMaxSize >	582
IRL::StolenGraph< NodeType, StolenGraphType >	
Class to inherit from that will utilize an already set up graph and mask some of its behaviors by shadowing in the NodeType class. In order to have a consistent interface with other graph classes, the NodeType class that inherits StolenGraph (which works due to CRTP) must implement a NodeType getNeighbor(const UnsignedIndex_t a_neighbor_index) const method See planar_reconstruction/localizer_link_from_localized_separator_link.h as an example	583
IRL::StoredCappedDodecahedron< VertexType >	584
IRL::StoredConcaveBox< VertexType >	585
IRL::StoredDodecahedron< VertexType >	585
IRL::StoredHexahedron< VertexType >	586
IRL::StoredPlaneOfExistence	586
IRL::StoredPolyhedron24< VertexType >	587
IRL::StoredRectangularCuboid< VertexType >	588
IRL::StoredTet< VertexType >	588
IRL::StoredTri< VertexType >	589
IRL::StoredVertexAccess< Derived, VertexType, kNumberOfVertices >	590
IRL::StrongWolfeConditionLineSearch	
Implementation of line search algorithm from "Line search algorithms with guaranteed sufficient decrease" by Jorge More and David Thuente, ACM Transactions on Mathematical Software, 1994. Some implementation details (such as the zoom function) are also taken from the book Numerical Optimization by Jorge Nocedal and Stephen Wright, 2006	591
f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_type	592
f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_type	593
f_tagged_accumvm_vm_class::tagged_accumvm_vm_type	594
IRL::TaggedAccumulatedListedVolumeMoments< VolumeMomentsType >	594
IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType >	595
IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType >::TaggedMoments	597
f_tet_class::tet_type	597

IRL::TetSpecialization< Derived, VertexType >	598
f_tri_class::tri_type	599
IRL::TriSpecialization< Derived, VertexType >	600
IRL::UnDirectedGraphNode< NodeType >	
Class to inherit from to provide linking between planes in a planar reconstruction to other reconstructions, where the linking exists PER PLANE	602
IRL::UnitQuaternion	
Unit quaternion to be used to perform rotations	603
f_serializer::unpackandstore	604
IRL::vector_sum< V1, V2 >	604
IRL::Vertex< VertexLocationType >	606
IRL::VertexList< VertexStorage, VertexType >	607
IRL::VertexLocation< PtType >	608
f_vman_class::vman_type	608
IRL::Volume	
A volume class which is just a double with special properties. Allows more general writing of functions	609
IRL::Volume2D_Functor	610
IRL::Volume3D_Functor	611
IRL::VolumeMoments	
Zeroeth (volume) and first order (centroid) geometric moments	611
IRL::VolumeMoments2D_Functor	614
IRL::VolumeMoments3D_Functor	615
IRL::VolumeMomentsAndDoubles< kArrayLength >	616
IRL::VolumeMomentsAndDoubles3D_Functor< kArrayLength >	618
IRL::VolumeMomentsAndNormal	
Class that contains VolumeMoments for a triangle and the triangle's normal	619
IRL::VolumeMomentsAndNormal2D_Functor	621
f_listedvm_vman_class::zeronormalcomponent	622
f_dividedpolygon_class::zeropolygon	622
f_polygon_class::zeropolygon	622

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

accumulate_wrapper.h	??
accumulated_listed_volume_moments.h	??
accumulated_volume_moments.h	??
accumulator.h	??
adverted_plane_reconstruction.h	??
bfgs.h	??
bisection.h	??
block_object_allocation.h	??
brents_method.h	??
byte_buffer.h	??
c_byte_buffer.h	??
c_capped_dodecahedron.h	??
c_capped_dodecahedron_doubles3.h	??
c_constants.h	623
c_cut_polygon.h	625
c_divided_polygon.h	??
c_dodecahedron.h	??
c_elvira_neighborhood.h	??
c_generic_cutting.h	627
c_geometric_cutting_helpers.h	??
c_listedvm_vman.h	??
c_localized_separator_link.h	??
c_localizer_link.h	??
c_localizers.h	630
c_ivira_neighborhood_rectangular_cuboid.h	??
c_object_allocation_server_localized_separator_link.h	??
c_object_allocation_server_localizer_link.h	??
c_object_allocation_server_planar_localizer.h	??
c_object_allocation_server_planar_separator.h	??
c_polygon.h	??
c_polyhedron24.h	??
c_polyhedron24_doubles3.h	??
c_r2p_neighborhood_rectangular_cuboid.h	??
c_reconstruction_interface.h	??
c_rectangular_cuboid.h	??

c_separated_volume_moments.h	??
c_separated_volume_moments_doubles3.h	??
c_separators.h	??
c_serializer.h	631
c_tagged_accumulated_listed_volume_moments_and_normal.h	??
c_tagged_accumulated_separated_volume_moments.h	??
c_tagged_accumulated_volume_moments.h	??
c_tet.h	??
c_tri.h	??
c_volume_fraction_matching.h	??
c_volume_moments_and_normal.h	??
capped_dodecahedron.h	??
cell_collection.h	??
cell_grouped_moments.h	??
chained_block_storage.h	??
class_classifications.h	??
collection.h	??
concave_box.h	??
constants.h	632
continue_dividing_volume.h	??
cut_polygon.h	??
cut_simplex.h	??
cut_simplex_drivers.h	??
decomposed_polytope_vertex_storage.h	??
default_cutting_method.h	??
defined_types.h	??
delayed_expandable_vertex_access.h	??
divided_polygon.h	??
dodecahedron.h	??
elvira.h	??
elvira_neighborhood.h	??
expandable_pt_list.h	??
f_bytebuffer_class.f90	
This file contains the Fortran interface for the ByteBuffer class	634
f_cappeddodecahedron_class.f90	
This file contains the Fortran interface for the CappedDodecahedron class	635
f_cappeddodecahedron_doubles3_class.f90	
This file contains the Fortran interface for the CappedDodecahedron_doubles3 class	636
f_constants.f90	
This file contains the Fortran interface to IRL functions that deal with setting constants	637
f_cutpolygon.f90	
This file deals with intersecting polygons and generating polygons corresponding to planar reconstructions	637
f_dividedpolygon_class.f90	
This file contains the Fortran interface for the DividedPolygon class	638
f_dodecahedron_class.f90	
This file contains the Fortran interface for the Dodecahedron class	640
f_geometriccuttinghelpers.f90	
This file provides access to helper functions often used during geometric cutting	640
f_getvolumemoments.f90	
This file deals with subdivinding and integrating volume moments for polyhedra	641
f_localizedseparatorlink_class.f90	
This file allows use of the IRL LocalizedSeparatorLink class through a fortran interface	642
f_localizerlink_class.f90	
This file allows use of the IRL LocalizerLink class through a fortran interface	643
f_objectallocationserver_localizedseparatorlink_class.f90	
This file allows use of the IRL ObjectAllocationServer<LocalizedSeparatorLink> class through a fortran interface	644

f_objectallocationserver_localizerlink_class.f90	This file allows use of the IRL ObjectAllocationServer<LocalizerLink> class through a fortran interface	645
f_objectallocationserver_planarlocalizer_class.f90	This file allows use of the IRL ObjectAllocationServer<PlanarLocalizer> class through a fortran interface	646
f_objectallocationserver_planarseparator_class.f90	This file allows use of the IRL ObjectAllocationServer<PlanarSeparator> class through a fortran interface	647
f_planarlocalizer_class.f90	This file allows use of the IRL PlanarLocalizer class through a fortran interface	647
f_planarseparator_class.f90	This file allows use of the IRL PlanarSeparator class through a fortran interface	648
f_polygon_class.f90	This file contains the Fortran interface for the Polygon class	649
f_polyhedron24_class.f90	This file contains the Fortran interface for the Polyhedron24 class	651
f_polyhedron24_doubles3_class.f90	This file contains the Fortran interface for the Polyhedron24_doubles3 class	652
f_r2pneighborhood_rectangularcuboid_class.f90	This file contains functions reproducing the functionality of the IRL class R2PNeighborhood←_RectangularCuboid. The purpose of this is to allow building the stencil through references to then supply to obtain a PlanarSeparator using the R2P method	653
f_rectangularcuboid_class.f90	This file contains the Fortran interface for the RectangularCuboid class	654
f_sepvm_class.f90	This file contains the Fortran interface for volume moments classes	655
f_sepvm_doubles3_class.f90	This file contains the Fortran interface for volume moments classes	656
f_serializer.f90	This file deals with serializing IRL class objects into byte buffers. This is usually done before parallel communication via MPI using MPI_BYTE	657
f_tagged_accumlistedvm_vman_class.f90	This file contains the Fortran interface for volume moments classes	657
f_tagged_accumvm_sepvm_class.f90	This file contains the Fortran interface for volume moments classes	658
f_tagged_accumvm_vm_class.f90	This file contains the Fortran interface for volume moments classes	660
f_tet_class.f90	This file contains the Fortran interface for the Tet class	661
f_tri_class.f90	This file contains the Fortran interface for the Tri class	662
f_vman_class.f90	This file contains the Fortran interface for volume moments classes	663
f_volumefractionmatching.f90	This file deals with setting the distances to each plane in a planar reconstruction to match a given volume fraction for the provided cell	664
general_polygon.h	??
general_polyhedron.h	??
generic_cutting.h	??
generic_cutting_definitions.h	??
geometric_cutting_helpers.h	??
geometry_type_traits.h	??
half_edge.h	??
half_edge_cutting.h	??
half_edge_cutting_drivers.h	??
half_edge_cutting_helpers.h	??
half_edge_cutting_initializer.h	??

half_edge_polygon.h	??
half_edge_polyhedron.h	??
half_edge_polytope.h	??
handle_enclosed_volume.h	??
helper.h	665
hexahedron.h	??
illinois.h	??
irl_fortran_interface.f90	
This file serves to provide a single include directive when using the IRL fortran interface	666
iterative_distance_solver.h	??
iterator_through_bracket_operator.h	??
joined_reconstructions.h	??
joined_reconstructions_to_group.h	??
k_means.h	??
levenberg_marquardt.h	??
listed_volume_moments.h	??
lister.h	??
localized_separator.h	??
localized_separator_group.h	??
localized_separator_group_link.h	??
localized_separator_link.h	??
localizer_link.h	??
localizer_link_from_localized_separator_link.h	??
lookup_tables.h	667
lvira_neighborhood.h	??
lvira_optimization.h	675
masked_localized_separator_link.h	??
math_vector.h	??
mof.h	??
moment_calculation_through_simplices.h	??
moments_type_traits.h	??
multimaterial_accumulated_volume_moments.h	??
mymath.h	677
new_pt_calculation_functors.h	??
normal.h	??
null_reconstruction.h	??
object_allocation_server.h	??
optimizers.h	??
partition_by_normal_vector.h	??
path_graph_node.h	??
planar_localizer.h	??
planar_reconstruction.h	??
planar_reconstruction_type_traits.h	??
planar_separator.h	??
planar_separator_path.h	??
planar_separator_path_group.h	??
plane.h	??
plane_distance.h	678
polygon.h	??
polygon_moments_calculation.h	??
polygon_storage_types.h	??
polyhedron_24.h	??
polyhedron_moments_calculation.h	??
proxy_vertex_access.h	??
pt.h	??
pt_list.h	??
pt_with_data.h	??
r2p_neighborhood.h	??

r2p_optimization.h	680
reconstruction_cleaning.h	??
reconstruction_interface.h	??
reconstruction_link.h	??
rectangular_cuboid.h	??
recursive_simplex_cutting_initializer.h	??
reference_frame.h	??
referred_to_plane_of_existence.h	??
restricted_localizer_link_from_localized_separator_link.h	??
restricted_masked_localized_separator_link.h	??
rotations.h	682
secant.h	??
segmented_decomposed_polytope.h	??
segmented_half_edge_polygon.h	??
segmented_half_edge_polyhedron.h	??
segmented_half_edge_polytope.h	??
self_expanding_collection.h	??
separated_volume_moments.h	??
serializer.h	??
SFINAE_boiler_plate.h	??
short_alloc.h	??
simplex_cutting.h	??
simplex_cutting_drivers.h	??
simplex_cutting_initializer.h	??
simplex_wrapper.h	??
small_vector.h	??
stack_vector.h	??
stolen_graph.h	??
stored_plane_of_existence.h	??
stored_vertex_access.h	??
tagged_accumulated_listed_volume_moments.h	??
tagged_accumulated_volume_moments.h	??
tet.h	??
tri.h	??
un_directed_graph_node.h	??
unit_quaternion.h	??
volume.h	??
volume_fraction_matching.h	??
volume_moments.h	??
volume_moments_and_doubles.h	??
volume_moments_and_normal.h	??

Chapter 5

Namespace Documentation

5.1 f_bytebuffer_class Module Reference

A fortran type class that allows the creation of IRL's ByteBuffer class along with enabling some of its methods.

Data Types

- type `bytebuffer_type`
- type `c_bytebuffer`
- interface `dataptr`
- interface `F_ByteBuffer_dataPtr`
- interface `F_ByteBuffer_delete`
- interface `F_ByteBuffer_getSize`
- interface `F_ByteBuffer_new`
- interface `F_ByteBuffer_resetBufferPointer`
- interface `F_ByteBuffer_setSize`
- interface `getcobject`
- interface `getsize`
- interface `new`
- interface `resetbufferpointer`
- interface `setsize`

Functions/Subroutines

- impure elemental subroutine `bytebuffer_class_delete` (this)
- subroutine `bytebuffer_class_new` (this)
- type(`c_bytebuffer`) function `bytebuffer_class_getcobject` (this)
- integer(irl_largeoffsetindex_t) function `bytebuffer_class_getsize` (this)
- subroutine `bytebuffer_class_setsize` (this, a_size)
- subroutine `bytebuffer_class_resetbufferpointer` (this)
- integer(irl_byte_t) function, dimension(:), pointer `bytebuffer_class_dataptr` (this)

5.1.1 Detailed Description

A fortran type class that allows the creation of IRL's ByteBuffer class along with enabling some of its methods.

5.2 f_cappeddodecahedron_class Module Reference

A fortran type class that allows the creation of IRL's CappedDodecahedron class along with enabling some of its methods.

Data Types

- interface `adjustcaptomatchvolume`
- type `c_cappeddodecahedron`
- type `cappeddodecahedron_type`
- interface `construct`
- interface `F_CappedDodecahedron_adjustCapToMatchVolume`
- interface `F_CappedDodecahedron_construct`
- interface `F_CappedDodecahedron_delete`
- interface `F_CappedDodecahedron_getBoundingPts`
- interface `F_CappedDodecahedron_getPt`
- interface `F_CappedDodecahedron_new`
- interface `getboundingpts`
- interface `getcobject`
- interface `getpt`
- interface `new`

Functions/Subroutines

- impure elemental subroutine `cappeddodecahedron_class_delete` (this)
- subroutine `cappeddodecahedron_class_new` (this)
- type(`c_cappeddodecahedron`) function `cappeddodecahedron_class_getcobject` (this)
- subroutine `cappeddodecahedron_class_construct` (this, a_dodecahedron)
- subroutine `cappeddodecahedron_class_adjustcaptomatchvolume` (this, a_correct_signed_volume)
- subroutine `cappeddodecahedron_class_getboundingpts` (this, a_lower_pt, a_upper_pt)
- real(irl_double) function, dimension(3) `cappeddodecahedron_class_getpt` (this, a_index)

5.2.1 Detailed Description

A fortran type class that allows the creation of IRL's CappedDodecahedron class along with enabling some of its methods.

5.3 f_cappeddodecahedron_doubles3_class Module Reference

A fortran type class that allows the creation of IRL's CappedDodecahedron_doubles3 class along with enabling some of its methods.

Data Types

- interface `adjustcaptomatchvolume`
- type `c_cappeddodecahedron_doubles3`
- type `cappeddodecahedron_doubles3_type`
- interface `construct`
- interface `F_CappedDodecahedron_doubles3_adjustCapToMatchVolume`
- interface `F_CappedDodecahedron_doubles3_construct`
- interface `F_CappedDodecahedron_doubles3_delete`
- interface `F_CappedDodecahedron_doubles3_getBoundingPts`
- interface `F_CappedDodecahedron_doubles3_getData`
- interface `F_CappedDodecahedron_doubles3_getPt`
- interface `F_CappedDodecahedron_doubles3_new`
- interface `F_CappedDodecahedron_doubles3_setData`
- interface `F_CappedDodecahedron_doubles3_setPt`
- interface `getboundingpts`
- interface `getcobject`
- interface `getdata`
- interface `getpt`
- interface `new`
- interface `setdata`
- interface `setpt`

Functions/Subroutines

- impure elemental subroutine `cappeddodecahedron_doubles3_class_delete` (this)
- subroutine `cappeddodecahedron_doubles3_class_new` (this)
- type(`c_cappeddodecahedron_doubles3`) function `cappeddodecahedron_doubles3_class_getcobject` (this)
- subroutine `cappeddodecahedron_doubles3_class_construct` (this, a_dodecahedron, a_attached_data)
- subroutine `cappeddodecahedron_doubles3_class_adjustcaptomatchvolume` (this, a_correct_signed←_volume)
- subroutine `cappeddodecahedron_doubles3_class_getboundingpts` (this, a_lower_pt, a_upper_pt)
- real(irl_double) function, dimension(3) `cappeddodecahedron_doubles3_class_getpt` (this, a_index)
- subroutine `cappeddodecahedron_doubles3_class_setpt` (this, a_index, a_pt)
- real(irl_double) function, dimension(3) `cappeddodecahedron_doubles3_class_getdata` (this, a_index)
- subroutine `cappeddodecahedron_doubles3_class_setdata` (this, a_index, a_data)

5.3.1 Detailed Description

A fortran type class that allows the creation of [IRL](#)'s CappedDodecahedron_doubles3 class along with enabling some of its methods.

5.4 f_constants Module Reference

This module contains mappings to the [IRL](#) C interface that deal with setting global constants that are used in the [IRL](#) library.

Data Types

- interface [F_Constants_setMinimumSurfaceAreaToTrack](#)
- interface [F_Constants_setMinimumVolumeToTrack](#)
- interface [F_Constants_setVolumeFractionBounds](#)
- interface [F_Constants_setVolumeFractionToleranceForDistanceFinding](#)

Functions/Subroutines

- subroutine **constants_setvolumefractionbounds** (a_VF_low)
- subroutine **constants_setvolumefractiontolerancefordistancefinding** (a_tolerance)
- subroutine **constants_setminimumvolumetotrack** (a_minimum_volume_to_track)
- subroutine **constants_setminimumsurfaceareatotrack** (a_minimum_surface_area_to_track)

5.4.1 Detailed Description

This module contains mappings to the [IRL](#) C interface that deal with setting global constants that are used in the [IRL](#) library.

5.5 f_cutpolygon Module Reference

This module contains mappings to the [IRL](#) C interface that deal with intersecting planes to generate polygons and creating polygons that are representative of planar reconstructions in given cells.

Data Types

- interface [F_getPlanePolygonFromReconstruction_RC_DivPoly](#)
- interface [F_getPlanePolygonFromReconstruction_RC_Poly](#)
- interface [F_getReconstructionSurfaceArea_RC](#)
- interface [getplanepolygonfromreconstruction](#)
- interface [getreconstructionsurfacearea](#)

Functions/Subroutines

- subroutine **getplanepolygonfromreconstruction_rc_poly** (a_rectangular_cuboid, a_planar_separator, a_plane_index, a_polygon)
- subroutine **getplanepolygonfromreconstruction_rc_divpoly** (a_rectangular_cuboid, a_planar_separator, a_plane_index, a_divided_polygon)
- real(irl_double) function **getreconstructionsurfacearea_rc** (a_rectangular_cuboid, a_planar_separator)

5.5.1 Detailed Description

This module contains mappings to the [IRL](#) C interface that deal with intersecting planes to generate polygons and creating polygons that are representative of planar reconstructions in given cells.

5.6 f_definedtypes Module Reference

This module contains mappings to the [IRL](#) C interface that deal with intersecting planes to generate polygons and creating polygons that are representative of planar reconstructions in given cells.

Variables

- integer, parameter `irl_unsignedindex_t` = 4
- integer, parameter `irl_signedindex_t` = 4
- integer, parameter `irl_largetoffsetindex_t` = 8
- integer, parameter `irl_byte_t` = 1
- integer, parameter `irl_double` = 8

5.6.1 Detailed Description

This module contains mappings to the [IRL](#) C interface that deal with intersecting planes to generate polygons and creating polygons that are representative of planar reconstructions in given cells.

5.7 f_dividedpolygon_class Module Reference

A fortran type class that allows the creation of [IRL](#)'s DividedPolygon class along with enabling some of its methods.

Data Types

- type `c_dividedpolygon`
- interface `calculateandsetplaneofexistence`
- interface `calculatenormal`
- interface `calculatesign`
- interface `calculatesurfacearea`
- interface `construct`
- interface `constructfrompolygon`
- type `dividedpolygon_type`
- interface `F_DividedPolygon_calculateAndSetPlaneOfExistence`
- interface `F_DividedPolygon_calculateNormal`
- interface `F_DividedPolygon_calculateSign`
- interface `F_DividedPolygon_calculateSurfaceArea`
- interface `F_DividedPolygon_construct`
- interface `F_DividedPolygon_constructFromPolygon`
- interface `F_DividedPolygon_delete`
- interface `F_DividedPolygon_getBoundingPts`
- interface `F_DividedPolygon_getLocalizer`
- interface `F_DividedPolygon_getNumberOfPts`
- interface `F_DividedPolygon_getNumberOfSimplicesInDecomposition`
- interface `F_DividedPolygon_getPlaneOfExistence`
- interface `F_DividedPolygon_getPt`
- interface `F_DividedPolygon_getSimplexFromDecomposition`
- interface `F_DividedPolygon_new`
- interface `F_DividedPolygon_printToScreen`

- interface `F_DividedPolygon_resetCentroid`
- interface `F_DividedPolygon_reversePtOrdering`
- interface `F_DividedPolygon_setPlaneOfExistence`
- interface `F_DividedPolygon_zeroPolygon`
- interface `getboundingpts`
- interface `getcobject`
- interface `getlocalizer`
- interface `getnumberofsimplicesindecomposition`
- interface `getnumberofvertices`
- interface `getplaneofexistence`
- interface `getpt`
- interface `getsimplexfromdecomposition`
- interface `new`
- interface `printtoscreen`
- interface `resetcentroid`
- interface `reverseptordering`
- interface `setplaneofexistence`
- interface `zeropolyon`

Functions/Subroutines

- subroutine `dividedpolygon_class_new` (this)
- impure elemental subroutine `dividedpolygon_class_delete` (this)
- type(`c_dividedpolygon`) function `dividedpolygon_class_getcobject` (this)
- subroutine `dividedpolygon_class_construct` (this, a_npts, a_pts)
- subroutine `dividedpolygon_class_constructfrompolygon` (this, a_polygon)
- subroutine `dividedpolygon_class_resetcentroid` (this)
- integer(irl_unsignedindex_t) function `dividedpolygon_class_getnumberofsimplicesindecomposition` (this)
- subroutine `dividedpolygon_class_getsimplexfromdecomposition` (this, a_tri_number_to_get, a_tri_in←_decomposition)
- real(irl_double) function, dimension(1:3) `dividedpolygon_class_calculatenormal` (this)
- subroutine `dividedpolygon_class_getlocalizer` (this, a_planar_localizer)
- subroutine `dividedpolygon_class_reverseptordering` (this)
- subroutine `dividedpolygon_class_getboundingpts` (this, a_lower_pt, a_upper_pt)
- integer(irl_unsignedindex_t) function `dividedpolygon_class_getnumberoftpts` (this)
- real(irl_double) function, dimension(3) `dividedpolygon_class_getpt` (this, a_index)
- subroutine `dividedpolygon_class_zeropolyon` (this)
- real(irl_double) function `dividedpolygon_class_calculatesurfacearea` (this)
- real(irl_double) function `dividedpolygon_class_calculatesign` (this)
- subroutine `dividedpolygon_class_setplaneofexistence` (this, a_plane)
- subroutine `dividedpolygon_class_calculateandsetplaneofexistence` (this)
- real(irl_double) function, dimension(4) `dividedpolygon_class_getplaneofexistence` (this)
- subroutine `dividedpolygon_class_printtoscreen` (this)

5.7.1 Detailed Description

A fortran type class that allows the creation of IRL's DividedPolygon class along with enabling some of its methods.

5.8 f_dodecahedron_class Module Reference

A fortran type class that allows the creation of IRL's Dodecahedron class along with enabling some of its methods.

Data Types

- type `c_dodecahedron`
- interface `construct`
- type `dodecahedron_type`
- interface `F_Dodecahedron_construct`
- interface `F_Dodecahedron_delete`
- interface `F_Dodecahedron_getBoundingPts`
- interface `F_Dodecahedron_new`
- interface `getboundingpts`
- interface `getcobject`
- interface `new`

Functions/Subroutines

- subroutine `dodecahedron_class_new` (this)
- impure elemental subroutine `dodecahedron_class_delete` (this)
- type(`c_dodecahedron`) function `dodecahedron_class_getcobject` (this)
- subroutine `dodecahedron_class_construct` (this, a_transported_cell)
- subroutine `dodecahedron_class_getboundingpts` (this, a_lower_pt, a_upper_pt)

5.8.1 Detailed Description

A fortran type class that allows the creation of IRL's Dodecahedron class along with enabling some of its methods.

5.9 f_elviraneighborhood_class Module Reference

A fortran type class to provide the functionality of ELVIRANeighborhood.

Data Types

- type `c_elviraneighborhood`
- type `elviraneighborhood_type`
- interface `F_ELVIRANeighborhood_delete`
- interface `F_ELVIRANeighborhood_new`
- interface `F_ELVIRANeighborhood_setMember`
- interface `F_ELVIRANeighborhood_setSize`
- interface `getcobject`
- interface `new`
- interface `setmember`
- interface `setsze`

Functions/Subroutines

- subroutine **elviraneighborhood_class_new** (this)
- impure elemental subroutine **elviraneighborhood_class_delete** (this)
- type([c_elviraneighborhood](#)) function **elviraneighborhood_class_getobject** (this)
- subroutine **elviraneighborhood_class_setsize** (this, a_size)
- subroutine **elviraneighborhood_class_setmember** (this, a_rectangular_cuboid, a_liquid_volume_fraction, i, j, k)

5.9.1 Detailed Description

A fortran type class to provide the functionality of ELVIRANeighborhood.

5.10 f_geometriccuttinghelpers Module Reference

This module contains mappings to the [IRL](#) C interface that provides access to functions often used to geoemtric cutting operations. See the C interface file src/c_interface/c_geometric_cutting_helpers.h for more information.

Data Types

- interface [F_isPtInternal_PL](#)
- interface [F_isPtInternal_PS](#)
- interface [isptinternal](#)

Functions/Subroutines

- logical(1) function [isptinternal_ps](#) (a_pt, a_separator)
- logical(1) function [isptinternal_pl](#) (a_pt, a_localizer)

5.10.1 Detailed Description

This module contains mappings to the [IRL](#) C interface that provides access to functions often used to geoemtric cutting operations. See the C interface file src/c_interface/c_geometric_cutting_helpers.h for more information.

5.11 f_getvolumemoments Module Reference

This module contains mappings to the [IRL](#) C interface that deal with intersecting polyhedron volumes and integrating these volumes to obtain volumetric moments.

Data Types

- interface [F_GNVM_CD_By_LSL_For_SVM](#)
- interface [F_GNVM_CD_By_LSL_For_TagAccumVM_SVM](#)
- interface [F_GNVM_CDWD3_By_LSL_For_SVMAD3](#)
- interface [F_GNVM_D_By_LSL_For_SVM](#)
- interface [F_GNVM_D_By_LSL_For_TagAccumVM_SVM](#)
- interface [F_GNVM_D_By_PS_For_SVM](#)
- interface [F_GNVM_P24_By_LSL_For_SVM](#)
- interface [F_GNVM_P24WD3_By_LSL_For_SVMAD3](#)
- interface [F_GNVM_Poly_By_PL_For_V](#)
- interface [F_GNVM_RC_By_PS_For_SVM](#)
- interface [F_GNVM_RC_By_PS_For_V](#)
- interface [F_GNVM_Tet_By_LSL_For_SVM](#)
- interface [F_GNVM_Tri_By_LL_For_TagAVM_VM](#)
- interface [F_GNVM_Tri_By_PL_For_V](#)
- interface [F_GVM_CD_By_LSL_For_SVM](#)
- interface [F_GVM_D_By_LSL_For_SVM](#)
- interface [F_GVM_P24_By_LSL_For_SVM](#)
- interface [F_GVM_setMethod](#)
- interface [F_GVM_Tri_By_LL_For_TagALVM_VMAN](#)
- interface [getnormalizedvolumemoments](#)
- interface [getvolumemoments](#)
- interface [getvolumemoments_setmethod](#)

Functions/Subroutines

- subroutine [gvm_setmethod](#) (a_cutting_method)
- subroutine [gnvm_d_by_lsl_for_svm](#) (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine [gnvm_cd_by_lsl_for_svm](#) (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine [gnvm_cdwd3_by_lsl_for_svmad3](#) (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine [gnvm_p24_by_lsl_for_svm](#) (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)
- subroutine [gnvm_p24wd3_by_lsl_for_svmad3](#) (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)
- subroutine [gvm_cd_by_lsl_for_svm](#) (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine [gvm_d_by_lsl_for_svm](#) (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine [gvm_p24_by_lsl_for_svm](#) (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)
- subroutine [gnvm_tet_by_lsl_for_svm](#) (a_tet, a_localized_separator_link, a_moments_to_return)
- subroutine [gnvm_rc_by_ps_for_v](#) (a_rectangular_cuboid, a_planar_separator, a_moments_to_return)
- subroutine [gnvm_d_by_ps_for_svm](#) (a_Dodecahedron, a_planar_separator, a_moments_to_return)
- subroutine [gnvm_cd_by_lsl_for_tagaccumvm_svm](#) (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine [gnvm_d_by_lsl_for_tagaccumvm_svm](#) (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine [gnvm_rc_by_ps_for_svm](#) (a_rectangular_cuboid, a_planar_separator, a_moments_to_return)
- subroutine [gnvm_tri_by_ll_for_tagavm_vm](#) (a_tri, a_localizer_link, a_moments_to_return)
- subroutine [gnvm_tri_by_pl_for_v](#) (a_tri, a_planar_localizer, a_moments_to_return)
- subroutine [gnvm_poly_by_pl_for_v](#) (a_polygon, a_planar_localizer, a_moments_to_return)
- subroutine [gvm_tri_by_ll_for_tagalvm_vman](#) (a_tri, a_localizer_link, a_moments_to_return)

5.11.1 Detailed Description

This module contains mappings to the [IRL](#) C interface that deal with intersecting polyhedron volumes and integrating these volumes to obtain volumetric moments.

5.12 f_listedvm_vman_class Module Reference

A fortran type class that allows the creation of [IRL](#)'s ListedVolumeMomentsM<VolumeMomentsAndNormal> class along with enabling some of its methods.

Data Types

- interface [append](#)
- type [c_listedvm_vman](#)
- interface [clear](#)
- interface [erase](#)
- interface [F_ListedVM_VMAN_append](#)
- interface [F_ListedVM_VMAN_clear](#)
- interface [F_ListedVM_VMAN_delete](#)
- interface [F_ListedVM_VMAN_erase](#)
- interface [F_ListedVM_VMAN_getMoments](#)
- interface [F_ListedVM_VMAN_getSize](#)
- interface [F_ListedVM_VMAN_new](#)
- interface [F_ListedVM_VMAN_zeroNormalComponent](#)
- interface [getcobject](#)
- interface [getmoments](#)
- interface [getsize](#)
- type [listedvm_vman_type](#)
- interface [new](#)
- interface [zeronormalcomponent](#)

Functions/Subroutines

- subroutine [listedvm_vman_class_new](#) (this)
- impure elemental subroutine [listedvm_vman_class_delete](#) (this)
- type([c_listedvm_vman](#)) function [listedvm_vman_class_getcobject](#) (this)
- subroutine [listedvm_vman_class_append](#) (this, a_other_list)
- subroutine [listedvm_vman_class_clear](#) (this)
- integer(irl_unsignedindex_t) function [listedvm_vman_class_getsize](#) (this)
- subroutine [listedvm_vman_class_getmoments](#) (this, a_index, a_moments)
- subroutine [listedvm_vman_class_zeronormalcomponent](#) (this, a_index)
- subroutine [listedvm_vman_class_erase](#) (this, a_index)

5.12.1 Detailed Description

A fortran type class that allows the creation of [IRL](#)'s ListedVolumeMomentsM<VolumeMomentsAndNormal> class along with enabling some of its methods.

5.13 f_localizedseparatorlink_class Module Reference

A fortran type class that allows the creation of IRL's LocalizedSeparatorLink class along with enabling some of its methods.

Data Types

- type [c_localizedseparatorlink](#)
- interface [F_LocalizedSeparatorLink_delete](#)
- interface [F_LocalizedSeparatorLink_getId](#)
- interface [F_LocalizedSeparatorLink_new](#)
- interface [F_LocalizedSeparatorLink_newFromObjectAllocationServer](#)
- interface [F_LocalizedSeparatorLink_setEdgeConnectivity](#)
- interface [F_LocalizedSeparatorLink_setEdgeConnectivityNull](#)
- interface [F_LocalizedSeparatorLink_setId](#)
- interface [getcobject](#)
- interface [getid](#)
- type [localizedseparatorlink_type](#)
- interface [new](#)
- interface [setedgeconnectivity](#)
- interface [setedgeconnectivitynull](#)
- interface [setid](#)

Functions/Subroutines

- subroutine [localizedseparatorlink_class_new](#) (this, a_planar_localizer, a_planar_separator)
- subroutine [localizedseparatorlink_class_newfromobjectallocationserver](#) (this, a_object_allocation_server, a_planar_localizer, a_planar_separator)
- impure elemental subroutine [localizedseparatorlink_class_delete](#) (this)
- type([c_localizedseparatorlink](#)) function [localizedseparatorlink_class_getcobject](#) (this)
- subroutine [localizedseparatorlink_class_setid](#) (this, a_id)
- integer(irl_unsignedindex_t) function [localizedseparatorlink_class_getid](#) (this)
- subroutine [localizedseparatorlink_class_setedgeconnectivity](#) (this, a_plane_index, a_neighboring_LocalizedSeparatorLink)
- subroutine [localizedseparatorlink_class_setedgeconnectivitynull](#) (this, a_plane_index)

5.13.1 Detailed Description

A fortran type class that allows the creation of IRL's LocalizedSeparatorLink class along with enabling some of its methods.

5.14 f_localizerlink_class Module Reference

A fortran type class that allows the creation of IRL's LocalizerLink class along with enabling some of its methods.

Data Types

- type `c_localizerlink`
- interface `F_LocalizerLink_delete`
- interface `F_LocalizerLink_getId`
- interface `F_LocalizerLink_new`
- interface `F_LocalizerLink_newFromObjectAllocationServer`
- interface `F_LocalizerLink_setEdgeConnectivity`
- interface `F_LocalizerLink_setEdgeConnectivityNull`
- interface `F_LocalizerLink_setId`
- interface `getcobject`
- interface `getid`
- type `localizerlink_type`
- interface `new`
- interface `setedgeconnectivity`
- interface `setedgeconnectivitynull`
- interface `setid`

Functions/Subroutines

- subroutine `localizerlink_class_new` (this, a_planar_localizer)
- subroutine `localizerlink_class_newfromobjectallocationserver` (this, a_object_allocation_server, a_<planar_localizer>)
- impure elemental subroutine `localizerlink_class_delete` (this)
- type(`c_localizerlink`) function `localizerlink_class_getcobject` (this)
- subroutine `localizerlink_class_setid` (this, a_id)
- integer(irl_unsignedindex_t) function `localizerlink_class_getid` (this)
- subroutine `localizerlink_class_setedgeconnectivity` (this, a_plane_index, a_neighboring_LocalizerLink)
- subroutine `localizerlink_class_setedgeconnectivitynull` (this, a_plane_index)

5.14.1 Detailed Description

A fortran type class that allows the creation of IRL's LocalizerLink class along with enabling some of its methods.

5.15 f_lviraneighborhood_rectangularcuboid_class Module Reference

A fortran type class to provide the functionality of LVIRANeighborhood.

Data Types

- interface `addmember`
- type `c_lviraneighborhood_rectangularcuboid`
- interface `emptyneighborhood`
- interface `F_LVIRANeighborhood_RectangularCuboid_addMember`
- interface `F_LVIRANeighborhood_RectangularCuboid_delete`
- interface `F_LVIRANeighborhood_RectangularCuboid_emptyNeighborhood`
- interface `F_LVIRANeighborhood_RectangularCuboid_new`
- interface `F_LVIRANeighborhood_RectangularCuboid_setCenterOfStencil`
- interface `F_LVIRANeighborhood_RectangularCuboid_setMember`
- interface `F_LVIRANeighborhood_RectangularCuboid_setSize`
- interface `getcobject`
- type `lviraneighborhood_rectangularcuboid_type`
- interface `new`
- interface `setcenterofstencil`
- interface `setmember`
- interface `setszie`

Functions/Subroutines

- subroutine **Iviraneighborhood_rectangularcuboid_class_new** (this)
- impure elemental subroutine **Iviraneighborhood_rectangularcuboid_class_delete** (this)
- type([c_Iviraneighborhood_rectangularcuboid](#)) function **Iviraneighborhood_rectangularcuboid_class_getcobject** (this)
- subroutine **Iviraneighborhood_rectangularcuboid_class_setsize** (this, a_size)
- subroutine **Iviraneighborhood_rectangularcuboid_class_setmember** (this, a_index, a_rectangular_cuboid, a_liquid_volume_fraction)
- subroutine **Iviraneighborhood_rectangularcuboid_class_addmember** (this, a_rectangular_cuboid, a_volume_fraction)
- subroutine **Iviraneighborhood_rectangularcuboid_class_emptyneighborhood** (this)
- subroutine **Iviraneighborhood_rectangularcuboid_class_setcenterofstencil** (this, a_center_cell_index)

5.15.1 Detailed Description

A fortran type class to provide the functionality of LVIRANeighborhood.

5.16 f_objectallocationserver_localizedseparatorlink_class Module Reference

A fortran type class that allows the creation of IRL's ObjectAllocationServer<LocalizedSeparatorLink> class along with enabling some of its methods.

Data Types

- type [c_objectallocationserver_localizedseparatorlink](#)
- interface [F_ObjectAllocationServer_LocalizedSeparatorLink_delete](#)
- interface [F_ObjectAllocationServer_LocalizedSeparatorLink_new](#)
- interface [getcobject](#)
- interface [new](#)
- type [objectallocationserver_localizedseparatorlink_type](#)

Functions/Subroutines

- subroutine **objectallocationserver_localizedseparatorlink_class_new** (this, a_number_to_allocate)
- impure elemental subroutine **objectallocationserver_localizedseparatorlink_class_delete** (this)
- type([c_objectallocationserver_localizedseparatorlink](#)) function **objectallocationserver_localizedseparatorlink_class_getcobject** (this)

5.16.1 Detailed Description

A fortran type class that allows the creation of IRL's ObjectAllocationServer<LocalizedSeparatorLink> class along with enabling some of its methods.

5.17 f_objectallocationserver_localizerlink_class Module Reference

A fortran type class that allows the creation of IRL's ObjectAllocationServer<LocalizerLink> class along with enabling some of its methods.

Data Types

- type [c_objectallocationserver_localizerlink](#)
- interface [F_ObjectAllocationServer_LocalizerLink_delete](#)
- interface [F_ObjectAllocationServer_LocalizerLink_new](#)
- interface [getcobject](#)
- interface [new](#)
- type [objectallocationserver_localizerlink_type](#)

Functions/Subroutines

- subroutine [objectallocationserver_localizerlink_class_new](#) (this, a_number_to_allocate)
- impure elemental subroutine [objectallocationserver_localizerlink_class_delete](#) (this)
- type([c_objectallocationserver_localizerlink](#)) function [objectallocationserver_localizerlink_class_getcobject](#) (this)

5.17.1 Detailed Description

A fortran type class that allows the creation of IRL's ObjectAllocationServer<LocalizerLink> class along with enabling some of its methods.

5.18 f_objectallocationserver_planarlocalizer_class Module Reference

A fortran type class that allows the creation of IRL's ObjectAllocationServer<PlanarLocalizer> class along with enabling some of its methods.

Data Types

- type [c_objectallocationserver_planarlocalizer](#)
- interface [F_ObjectAllocationServer_PlanarLocalizer_delete](#)
- interface [F_ObjectAllocationServer_PlanarLocalizer_new](#)
- interface [getcobject](#)
- interface [new](#)
- type [objectallocationserver_planarlocalizer_type](#)

Functions/Subroutines

- subroutine [objectallocationserver_planarlocalizer_class_new](#) (this, a_number_to_allocate)
- impure elemental subroutine [objectallocationserver_planarlocalizer_class_delete](#) (this)
- type([c_objectallocationserver_planarlocalizer](#)) function [objectallocationserver_planarlocalizer_class_getcobject](#) (this)

5.18.1 Detailed Description

A fortran type class that allows the creation of IRL's ObjectAllocationServer<PlanarLocalizer> class along with enabling some of its methods.

5.19 f_objectallocationserver_planarseparator_class Module Reference

A fortran type class that allows the creation of IRL's ObjectAllocationServer<PlanarSeparator> class along with enabling some of its methods.

Data Types

- type [c_objectallocationserver_planarseparator](#)
- interface [F_ObjectAllocationServer_PlanarSeparator_delete](#)
- interface [F_ObjectAllocationServer_PlanarSeparator_new](#)
- interface [getcobject](#)
- interface [new](#)
- type [objectallocationserver_planarseparator_type](#)

Functions/Subroutines

- subroutine [objectallocationserver_planarseparator_class_new](#) (this, a_number_to_allocate)
- impure elemental subroutine [objectallocationserver_planarseparator_class_delete](#) (this)
- type([c_objectallocationserver_planarseparator](#)) function [objectallocationserver_planarseparator_class_getcobject](#) (this)

5.19.1 Detailed Description

A fortran type class that allows the creation of IRL's ObjectAllocationServer<PlanarSeparator> class along with enabling some of its methods.

5.20 f_planarlocalizer_class Module Reference

A fortran type class that allows the creation of IRL's PlanarLocalizer class along with enabling some of its methods.

Data Types

- interface [addplane](#)
- type [c_planarlocalizer](#)
- interface [F_PlanarLocalizer_addPlane](#)
- interface [F_PlanarLocalizer_delete](#)
- interface [F_PlanarLocalizer_new](#)
- interface [F_PlanarLocalizer_newFromObjectAllocationServer](#)
- interface [F_PlanarLocalizer_printToScreen](#)
- interface [F_PlanarLocalizer_setFromRectangularCuboid](#)
- interface [F_PlanarLocalizer_setNumberOfPlanes](#)
- interface [F_PlanarLocalizer_setPlane](#)
- interface [getcobject](#)
- interface [new](#)
- type [planarlocalizer_type](#)
- interface [printtoscreen](#)
- interface [setfromrectangularcuboid](#)
- interface [setnumberofplanes](#)
- interface [setplane](#)

Functions/Subroutines

- subroutine **planarlocalizer_class_new** (this)
- subroutine **planarlocalizer_class_newfromobjectallocationserver** (this, a_object_allocation_server)
- impure elemental subroutine **planarlocalizer_class_delete** (this)
- type([c_planarlocalizer](#)) function **planarlocalizer_class_getcobject** (this)
- subroutine **planarlocalizer_class_addplane** (this, a_normal, a_distance)
- subroutine **planarlocalizer_class_setnumberofplanes** (this, a_number_to_set)
- subroutine **planarlocalizer_class_setplane** (this, a_plane_index_to_set, a_normal, a_distance)
- subroutine **planarlocalizer_class_setfromrectangularcuboid** (this, a_lower_pt, a_upper_pt)
- subroutine **planarlocalizer_class_printtoscreen** (this)

5.20.1 Detailed Description

A fortran type class that allows the creation of [IRL](#)'s PlanarLocalizer class along with enabling some of its methods.

5.21 f_planarseparator_class Module Reference

A fortran type class that allows the creation of [IRL](#)'s PlanarSeparator class along with enabling some of its methods.

Data Types

- interface [addplane](#)
- type [c_planarseparator](#)
- interface [copy](#)
- interface [F_PlanarSeparator_addPlane](#)
- interface [F_PlanarSeparator_copy](#)
- interface [F_PlanarSeparator_delete](#)
- interface [F_PlanarSeparator_getNumberOfPlanes](#)
- interface [F_PlanarSeparator_getPlane](#)
- interface [F_PlanarSeparator_isFlipped](#)
- interface [F_PlanarSeparator_new](#)
- interface [F_PlanarSeparator_newFromObjectAllocationServer](#)
- interface [F_PlanarSeparator_printToScreen](#)
- interface [F_PlanarSeparator_setNumberOfPlanes](#)
- interface [F_PlanarSeparator_setPlane](#)
- interface [getcobject](#)
- interface [getnumberofplanes](#)
- interface [getplane](#)
- interface [isflipped](#)
- interface [new](#)
- type [planarseparator_type](#)
- interface [printtoscreen](#)
- interface [setnumberofplanes](#)
- interface [setplane](#)

Functions/Subroutines

- subroutine **planarseparator_class_new** (this)
- subroutine **planarseparator_class_newfromobjectallocationserver** (this, a_object_allocation_server)
- impure elemental subroutine **planarseparator_class_delete** (this)
- type([c_planarseparator](#)) function **planarseparator_class_getcobject** (this)
- subroutine **planarseparator_class_addplane** (this, a_normal, a_distance)
- subroutine **planarseparator_class_setnumberofplanes** (this, a_number_to_set)
- subroutine **planarseparator_class_setplane** (this, a_plane_index_to_set, a_normal, a_distance)
- subroutine **planarseparator_class_copy** (this, a_other_PlanarSeparator)
- integer(irl_unsignedindex_t) function **planarseparator_class_getnumberofplanes** (this)
- real(irl_double) function, dimension(4) **planarseparator_class_getplane** (this, a_index)
- logical(1) function **planarseparator_class_isflipped** (this)
- subroutine **planarseparator_class_printtoscreen** (this)

5.21.1 Detailed Description

A fortran type class that allows the creation of [IRL](#)'s PlanarSeparator class along with enabling some of its methods.

5.22 f_polygon_class Module Reference

A fortran type class that allows the creation of [IRL](#)'s Polygon class along with enabling some of its methods.

Data Types

- type [c_polygon](#)
- interface [calculateandsetplaneofexistence](#)
- interface [calculatecentroid](#)
- interface [calculatenarestptonsurface](#)
- interface [calculatenormal](#)
- interface [calculatesign](#)
- interface [calculatevolume](#)
- interface [construct](#)
- interface [F_Polygon_calculateAndSetPlaneOfExistence](#)
- interface [F_Polygon_calculateCentroid](#)
- interface [F_Polygon_calculateNearestPtOnSurface](#)
- interface [F_Polygon_calculateNormal](#)
- interface [F_Polygon_calculateSign](#)
- interface [F_Polygon_calculateVolume](#)
- interface [F_Polygon_construct](#)
- interface [F_Polygon_delete](#)
- interface [F_Polygon_getBoundingPts](#)
- interface [F_Polygon_getLocalizer](#)
- interface [F_Polygon_getNumberOfPts](#)
- interface [F_Polygon_getNumberOfSimplicesInDecomposition](#)
- interface [F_Polygon_getPlaneOfExistence](#)
- interface [F_Polygon_getPt](#)
- interface [F_Polygon_getSimplexFromDecomposition](#)
- interface [F_Polygon_new](#)

- interface `F_Polygon_printToScreen`
- interface `F_Polygon_reversePtOrdering`
- interface `F_Polygon_setPlaneOfExistence`
- interface `F_Polygon_zeroPolygon`
- interface `getboundingpts`
- interface `getcobject`
- interface `getlocalizer`
- interface `getnumberofsimpllicesindecomposition`
- interface `getnumberofvertices`
- interface `getplaneofexistence`
- interface `getpt`
- interface `getsimplexfromdecomposition`
- interface `new`
- type `polygon_type`
- interface `printtoscreen`
- interface `reverseptordering`
- interface `setplaneofexistence`
- interface `zeropolyon`

Functions/Subroutines

- subroutine `polygon_class_new` (this)
- impure elemental subroutine `polygon_class_delete` (this)
- type(`c_polygon`) function `polygon_class_getcobject` (this)
- subroutine `polygon_class_construct` (this, a_npts, a_pts)
- real(irl_double) function, dimension(1:3) `polygon_class_calculatenormal` (this)
- subroutine `polygon_class_getlocalizer` (this, a_planar_localizer)
- subroutine `polygon_class_reverseptordering` (this)
- subroutine `polygon_class_getboundingpts` (this, a_lower_pt, a_upper_pt)
- integer(irl_unsignedindex_t) function `polygon_class_getnumberoftpts` (this)
- real(irl_double) function, dimension(3) `polygon_class_getpt` (this, a_index)
- integer(irl_unsignedindex_t) function `polygon_class_getnumberofsimpllicesindecomposition` (this)
- subroutine `polygon_class_getsimplexfromdecomposition` (this, a_tri_number_to_get, a_tri_in← decomposition)
- subroutine `polygon_class_zeropolyon` (this)
- real(irl_double) function, dimension(3) `polygon_class_calculatenearestptonsurface` (this, a_pt)
- real(irl_double) function `polygon_class_calculatevolume` (this)
- real(irl_double) function `polygon_class_calculatesign` (this)
- subroutine `polygon_class_setplaneofexistence` (this, a_plane)
- subroutine `polygon_class_calculateandsetplaneofexistence` (this)
- real(irl_double) function, dimension(4) `polygon_class_getplaneofexistence` (this)
- real(irl_double) function, dimension(3) `polygon_class_calculatecentroid` (this)
- subroutine `polygon_class_printtoscreen` (this)

5.22.1 Detailed Description

A fortran type class that allows the creation of IRL's Polygon class along with enabling some of its methods.

5.23 f_polyhedron24_class Module Reference

A fortran type class that allows the creation of IRL's Polyhedron24 class along with enabling some of its methods.

Data Types

- interface [adjustcaptomatchvolume](#)
- type [c_polyhedron24](#)
- interface [construct](#)
- interface [F_Polyhedron24_adjustCapToMatchVolume](#)
- interface [F_Polyhedron24_construct](#)
- interface [F_Polyhedron24_delete](#)
- interface [F_Polyhedron24_getBoundingPts](#)
- interface [F_Polyhedron24_getPt](#)
- interface [F_Polyhedron24_new](#)
- interface [F_Polyhedron24_setPt](#)
- interface [getboundingpts](#)
- interface [getcobject](#)
- interface [getpt](#)
- interface [new](#)
- type [polyhedron24_type](#)
- interface [setpt](#)

Functions/Subroutines

- subroutine **[polyhedron24_class_new](#)** (this)
- impure elemental subroutine **[polyhedron24_class_delete](#)** (this)
- type([c_polyhedron24](#)) function **[polyhedron24_class_getcobject](#)** (this)
- subroutine **[polyhedron24_class_construct](#)** (this, a_polyhedron24)
- subroutine **[polyhedron24_class_adjustcaptomatchvolume](#)** (this, a_correct_signed_volume)
- subroutine **[polyhedron24_class_getboundingpts](#)** (this, a_lower_pt, a_upper_pt)
- real(irl_double) function, dimension(3) **[polyhedron24_class_getpt](#)** (this, a_index)
- subroutine **[polyhedron24_class_setpt](#)** (this, a_index, a_pt)

5.23.1 Detailed Description

A fortran type class that allows the creation of IRL's Polyhedron24 class along with enabling some of its methods.

5.24 f_polyhedron24_doubles3_class Module Reference

A fortran type class that allows the creation of IRL's Polyhedron24_doubles3 class along with enabling some of its methods.

Data Types

- interface `adjustcaptomatchvolume`
- type `c_polyhedron24_doubles3`
- interface `construct`
- interface `F_Polyhedron24_doubles3_adjustCapToMatchVolume`
- interface `F_Polyhedron24_doubles3_construct`
- interface `F_Polyhedron24_doubles3_delete`
- interface `F_Polyhedron24_doubles3_getBoundingPts`
- interface `F_Polyhedron24_doubles3_getData`
- interface `F_Polyhedron24_doubles3_getPt`
- interface `F_Polyhedron24_doubles3_new`
- interface `F_Polyhedron24_doubles3_setData`
- interface `F_Polyhedron24_doubles3_setPt`
- interface `getboundingpts`
- interface `getcobject`
- interface `getdata`
- interface `getpt`
- interface `new`
- type `polyhedron24_doubles3_type`
- interface `setdata`
- interface `setpt`

Functions/Subroutines

- subroutine `polyhedron24_doubles3_class_new` (this)
- impure elemental subroutine `polyhedron24_doubles3_class_delete` (this)
- type(`c_polyhedron24_doubles3`) function `polyhedron24_doubles3_class_getcobject` (this)
- subroutine `polyhedron24_doubles3_class_construct` (this, a_polyhedron24, a_data)
- subroutine `polyhedron24_doubles3_class_adjustcaptomatchvolume` (this, a_correct_signed_volume)
- subroutine `polyhedron24_doubles3_class_getboundingpts` (this, a_lower_pt, a_upper_pt)
- real(irl_double) function, dimension(3) `polyhedron24_doubles3_class_getpt` (this, a_index)
- subroutine `polyhedron24_doubles3_class_setpt` (this, a_index, a_pt)
- real(irl_double) function, dimension(3) `polyhedron24_doubles3_class_getdata` (this, a_index)
- subroutine `polyhedron24_doubles3_class_setdata` (this, a_index, a_data)

5.24.1 Detailed Description

A fortran type class that allows the creation of IRL's Polyhedron24_doubles3 class along with enabling some of its methods.

5.25 f_r2pneighborhood_rectangularcuboid_class Module Reference

A fortran type class to provide the functionality of R2PNeighborhood_RectangularCuboid.

Data Types

- interface `addmember`
- type `c_r2pneighborhood_rectangularcuboid`
- interface `emptyneighborhood`
- interface `F_R2PNeighborhood_RectangularCuboid_addMember`
- interface `F_R2PNeighborhood_RectangularCuboid_delete`
- interface `F_R2PNeighborhood_RectangularCuboid_emptyNeighborhood`
- interface `F_R2PNeighborhood_RectangularCuboid_new`
- interface `F_R2PNeighborhood_RectangularCuboid_setCenterOfStencil`
- interface `F_R2PNeighborhood_RectangularCuboid_setMember`
- interface `F_R2PNeighborhood_RectangularCuboid_setSize`
- interface `F_R2PNeighborhood_RectangularCuboid_setSurfaceArea`
- interface `getcobject`
- interface `new`
- type `r2pneighborhood_rectangularcuboid_type`
- interface `setcenterofstencil`
- interface `setmember`
- interface `setszie`
- interface `setsurfacearea`

Functions/Subroutines

- subroutine `r2pneighborhood_rectangularcuboid_class_new` (this)
- impure elemental subroutine `r2pneighborhood_rectangularcuboid_class_delete` (this)
- type(`c_r2pneighborhood_rectangularcuboid`) function `r2pneighborhood_rectangularcuboid_class_getcobject` (this)
- subroutine `r2pneighborhood_rectangularcuboid_class_setsize` (this, a_size)
- subroutine `r2pneighborhood_rectangularcuboid_class_setmember` (this, a_rectangular_cuboid, a_separated_volume_moments, a_index)
- subroutine `r2pneighborhood_rectangularcuboid_class_addmember` (this, a_rectangular_cuboid, a_separated_volume_moments)
- subroutine `r2pneighborhood_rectangularcuboid_class_emptyneighborhood` (this)
- subroutine `r2pneighborhood_rectangularcuboid_class_setcenterofstencil` (this, a_center_cell_index)
- subroutine `r2pneighborhood_rectangularcuboid_class_setsurfacearea` (this, a_surface_area)

5.25.1 Detailed Description

A fortran type class to provide the functionality of R2PNeighborhood_RectangularCuboid.

5.26 f_reconstructioninterface Module Reference

This module contains interface reconstruction methods that can be used to obtain PlanarSeparators. The requirements to use each type of reconstruction are different. Please consult the documentation and examples before using a specific reconstruction type.

Data Types

- interface `F_reconstructionWithAdvectedNormals_ListedVM_VMAN_RC`
- interface `F_reconstructionWithAdvectedNormalsDebug_ListedVM_VMAN_RC`
- interface `F_reconstructionWithELVIRA2D`
- interface `F_reconstructionWithELVIRA3D`
- interface `F_reconstructionWithLVIRA2D_RC`
- interface `F_reconstructionWithLVIRA3D_RC`
- interface `F_reconstructionWithMOF2D_RectangularCuboid`
- interface `F_reconstructionWithMOF2D_Tri`
- interface `F_reconstructionWithMOF2DGiveWeights_RectangularCuboid`
- interface `F_reconstructionWithMOF2DGiveWeights_Tri`
- interface `F_reconstructionWithMOF3D_RectangularCuboid`
- interface `F_reconstructionWithMOF3D_Tet`
- interface `F_reconstructionWithMOF3DGiveWeights_RectangularCuboid`
- interface `F_reconstructionWithMOF3DGiveWeights_Tet`
- interface `F_reconstructionWithR2P2D_RC`
- interface `F_reconstructionWithR2P2DDebug_RC`
- interface `F_reconstructionWithR2P3D_RC`
- interface `F_reconstructionWithR2P3DDebug_RC`
- interface `reconstructionwithadvectedenormals`
- interface `reconstructionwithadvectedenormalsdebug`
- interface `reconstructionwithlvira2d`
- interface `reconstructionwithlvira3d`
- interface `reconstructionwithmof2d`
- interface `reconstructionwithmof3d`
- interface `reconstructionwithr2p2d`
- interface `reconstructionwithr2p2ddebug`
- interface `reconstructionwithr2p3d`
- interface `reconstructionwithr2p3ddebug`

Functions/Subroutines

- subroutine `reconstructionwithelvira2d` (`a_elvira_neighborhood`, `a_planar_separator`)
- subroutine `reconstructionwithelvira3d` (`a_elvira_neighborhood`, `a_planar_separator`)
- subroutine `reconstructionwithmof2d_rectangularcuboid` (`a_rectangular_cuboid`, `a_separated_volume_moments`, `a_planar_separator`)
- subroutine `reconstructionwithmof3d_rectangularcuboid` (`a_rectangular_cuboid`, `a_separated_volume_moments`, `a_planar_separator`)
- subroutine `reconstructionwithmof2dgiveweights_rectangularcuboid` (`a_rectangular_cuboid`, `a_separated_volume_moments`, `a_internal_weight`, `a_external_weight`, `a_planar_separator`)
- subroutine `reconstructionwithmof3dgiveweights_rectangularcuboid` (`a_rectangular_cuboid`, `a_separated_volume_moments`, `a_internal_weight`, `a_external_weight`, `a_planar_separator`)
- subroutine `reconstructionwithmof2d_tri` (`a_tri`, `a_separated_volume_moments`, `a_planar_separator`)
- subroutine `reconstructionwithmof2dgiveweights_tri` (`a_tri`, `a_separated_volume_moments`, `a_internal_weight`, `a_external_weight`, `a_planar_separator`)
- subroutine `reconstructionwithmof3d_tet` (`a_tet`, `a_separated_volume_moments`, `a_planar_separator`)
- subroutine `reconstructionwithmof3dgiveweights_tet` (`a_tet`, `a_separated_volume_moments`, `a_internal_weight`, `a_external_weight`, `a_planar_separator`)
- subroutine `reconstructionwithadvectedenormals_listedvm_vman_rc` (`a_volume_moments_list`, `a_neighborhood`, `a_two_plane_threshold`, `a_planar_separator`)
- subroutine `reconstructionwithadvectedenormalsdebug_listedvm_vman_rc` (`a_volume_moments_list`, `a_neighborhood`, `a_two_plane_threshold`, `a_planar_separator`)
- subroutine `reconstructionwithr2p2d_rc` (`a_neighborhood`, `a_planar_separator`)
- subroutine `reconstructionwithr2p3d_rc` (`a_neighborhood`, `a_planar_separator`)
- subroutine `reconstructionwithr2p2ddebug_rc` (`a_neighborhood`, `a_planar_separator`)
- subroutine `reconstructionwithr2p3ddebug_rc` (`a_neighborhood`, `a_planar_separator`)
- subroutine `reconstructionwithlvira2d_rc` (`a_neighborhood`, `a_planar_separator`)
- subroutine `reconstructionwithlvira3d_rc` (`a_neighborhood`, `a_planar_separator`)

5.26.1 Detailed Description

This module contains interface reconstruction methods that can be used to obtain PlanarSeparators. The requirements to use each type of reconstruction are different. Please consult the documentation and examples before using a specific reconstruction type.

5.27 f_rectangularcuboid_class Module Reference

A fortran type class that allows the creation of IRL's RectangularCuboid class along with enabling some of its methods.

Data Types

- type `c_rectangularcuboid`
- interface `calculatetable`
- interface `construct`
- interface `construct_2pt`
- interface `F_RectangularCuboid_calculateVolume`
- interface `F_RectangularCuboid_construct`
- interface `F_RectangularCuboid_construct_2pt`
- interface `F_RectangularCuboid_delete`
- interface `F_RectangularCuboid_getBoundingPts`
- interface `F_RectangularCuboid_new`
- interface `getboundingpts`
- interface `getcobject`
- interface `new`
- type `rectangularcuboid_type`

Functions/Subroutines

- subroutine `rectangularcuboid_class_new` (this)
- impure elemental subroutine `rectangularcuboid_class_delete` (this)
- type(`c_rectangularcuboid`) function `rectangularcuboid_class_getcobject` (this)
- subroutine `rectangularcuboid_class_construct` (this, a_transported_cell)
- subroutine `rectangularcuboid_class_construct_2pt` (this, a_lower_pt, a_upper_pt)
- real(irl_double) function `rectangularcuboid_class_calculatetable` (this)
- subroutine `rectangularcuboid_class_getboundingpts` (this, a_lower_pt, a_upper_pt)

5.27.1 Detailed Description

A fortran type class that allows the creation of IRL's RectangularCuboid class along with enabling some of its methods.

5.28 f_sepvm_class Module Reference

A fortran type class that allows the creation of IRL's SeparatedMoments<VolumeMoments> class along with enabling some of its methods.

Data Types

- type `c_sepvm`
- interface `construct`
- interface `F_SepVM_construct`
- interface `F_SepVM_delete`
- interface `F_SepVM_getCentroid`
- interface `F_SepVM_getCentroidPtr`
- interface `F_SepVM_getVolume`
- interface `F_SepVM_getVolumePtr`
- interface `F_SepVM_multiplyByVolume`
- interface `F_SepVM_new`
- interface `F_SepVM_normalizeByVolume`
- interface `getcentroid`
- interface `getcentroidptr`
- interface `getcobject`
- interface `getvolume`
- interface `getvolumeptr`
- interface `multiplybyvolume`
- interface `new`
- interface `normalizebyvolume`
- type `sepvm_type`

Functions/Subroutines

- subroutine `sepvm_class_new` (this)
- impure elemental subroutine `sepvm_class_delete` (this)
- type(`c_sepvm`) function `sepvm_class_getcobject` (this)
- subroutine `sepvm_class_construct` (this, a_moments_list)
- subroutine `sepvm_class_normalizebyvolume` (this)
- subroutine `sepvm_class_multiplybyvolume` (this)
- real(irl_double) function `sepvm_class_getvolume` (this, a_index)
- real(irl_double) function, dimension(3) `sepvm_class_getcentroid` (this, a_index)
- real(irl_double) function, pointer `sepvm_class_getvolumeptr` (this, a_index)
- real(irl_double) function, dimension(:), pointer `sepvm_class_getcentroidptr` (this, a_index)

5.28.1 Detailed Description

A fortran type class that allows the creation of `IRL`'s `SeparatedMoments<VolumeMoments>` class along with enabling some of its methods.

5.29 f_sepvm_doubles3_class Module Reference

A fortran type class that allows the creation of `IRL`'s `SeparatedMoments<VolumeMoments>` class along with enabling some of its methods.

Data Types

- type `c_sepvm_doubles3`
- interface `F_SepVM_doubles3_delete`
- interface `F_SepVM_doubles3_getCentroid`
- interface `F_SepVM_doubles3_getCentroidPtr`
- interface `F_SepVM_doubles3_getData`
- interface `F_SepVM_doubles3_getVolume`
- interface `F_SepVM_doubles3_getVolumePtr`
- interface `F_SepVM_doubles3_multiplyByVolume`
- interface `F_SepVM_doubles3_new`
- interface `F_SepVM_doubles3_normalizeByVolume`
- interface `getcentroid`
- interface `getcentroidptr`
- interface `getcobject`
- interface `getdata`
- interface `getvolume`
- interface `getvolumeptr`
- interface `multiplybyvolume`
- interface `new`
- interface `normalizebyvolume`
- type `sepvm_doubles3_type`

Functions/Subroutines

- subroutine `sepvm_doubles3_class_new` (this)
- impure elemental subroutine `sepvm_doubles3_class_delete` (this)
- type(`c_sepvm_doubles3`) function `sepvm_doubles3_class_getcobject` (this)
- subroutine `sepvm_doubles3_class_normalizebyvolume` (this)
- subroutine `sepvm_doubles3_class_multiplybyvolume` (this)
- real(irl_double) function `sepvm_doubles3_class_getvolume` (this, a_index)
- real(irl_double) function, dimension(3) `sepvm_doubles3_class_getcentroid` (this, a_index)
- real(irl_double) function, dimension(3) `sepvm_doubles3_class_getdata` (this, a_index)
- real(irl_double) function, pointer `sepvm_doubles3_class_getvolumeptr` (this, a_index)
- real(irl_double) function, dimension(:), pointer `sepvm_doubles3_class_getcentroidptr` (this, a_index)

5.29.1 Detailed Description

A fortran type class that allows the creation of `IRL`'s `SeparatedMoments<VolumeMoments>` class along with enabling some of its methods.

5.30 f_serializer Module Reference

This module contains mappings to the `IRL` C interface that deal with serializing `IRL` class objects into an array of bytes and packing them into a byte buffer.

Data Types

- interface `F_Serializer_serializeAndPack_PlanarSeparator_ByteBuffer`
- interface `F_Serializer_unpackAndStore_PlanarSeparator_ByteBuffer`
- interface `serializeandpack`
- interface `unpackandstore`

Functions/Subroutines

- subroutine `serializeandpack_planarseparator_bytebuffer` (`a_separator, a_byte_buffer`)
- subroutine `unpackandstore_planarseparator_bytebuffer` (`a_separator, a_byte_buffer`)

5.30.1 Detailed Description

This module contains mappings to the `IRL` C interface that deal with serializing `IRL` class objects into an array of bytes and packing them into a byte buffer.

5.31 `f_tagged_accumlistedvm_vman_class` Module Reference

A fortran type class that allows the creation of `IRL`'s `TaggedAccumulatedListedVolumeMomentsM<Volume,MomentsAndNormal>` class along with enabling some of its methods.

Data Types

- interface `append`
- type `c_tagged_accumlistedvm_vman`
- interface `clear`
- interface `F_Tagged_AccumListedVM_VMAN_append`
- interface `F_Tagged_AccumListedVM_VMAN_clear`
- interface `F_Tagged_AccumListedVM_VMAN_delete`
- interface `F_Tagged_AccumListedVM_VMAN_getListAtIndex`
- interface `F_Tagged_AccumListedVM_VMAN_getSize`
- interface `F_Tagged_AccumListedVM_VMAN_getTagForIndex`
- interface `F_Tagged_AccumListedVM_VMAN_new`
- interface `getcobject`
- interface `getlistatindex`
- interface `getsize`
- interface `gettagforindex`
- interface `new`
- type `tagged_accumlistedvm_vman_type`

Functions/Subroutines

- subroutine `tagged_accumlistedvm_vman_class_new` (`this`)
- impure elemental subroutine `tagged_accumlistedvm_vman_class_delete` (`this`)
- type(`c_tagged_accumlistedvm_vman`) function `tagged_accumlistedvm_vman_class_getcobject` (`this`)
- subroutine `tagged_accumlistedvm_vman_class_getlistatindex` (`this, a_index, a_other_list`)
- subroutine `tagged_accumlistedvm_vman_class_append` (`this, a_other_list`)
- subroutine `tagged_accumlistedvm_vman_class_clear` (`this`)
- integer(`irl_unsignedindex_t`) function `tagged_accumlistedvm_vman_class_getsize` (`this`)
- integer(`irl_unsignedindex_t`) function `tagged_accumlistedvm_vman_class_gettagforindex` (`this, a_index`)

5.31.1 Detailed Description

A fortran type class that allows the creation of IRL's TaggedAccumulatedListedVolumeMomentsM<Volume← MomentsAndNormal> class along with enabling some of its methods.

5.32 f_tagged_accumvm_sepvm_class Module Reference

A fortran type class that allows the creation of IRL's AccumulatedVolumeMomentsM<SeparatedMoments<← VolumeMoments>> class along with enabling some of its methods.

Data Types

- type [c_tagged_accumvm_sepvm](#)
- interface [F_Tagged_AccumVM_SepVM_delete](#)
- interface [F_Tagged_AccumVM_SepVM_getCentroidAtIndex](#)
- interface [F_Tagged_AccumVM_SepVM_getCentroidAtTag](#)
- interface [F_Tagged_AccumVM_SepVM_getCentroidPtrAtIndex](#)
- interface [F_Tagged_AccumVM_SepVM_getSize](#)
- interface [F_Tagged_AccumVM_SepVM_getTagForIndex](#)
- interface [F_Tagged_AccumVM_SepVM_getVolumeAtIndex](#)
- interface [F_Tagged_AccumVM_SepVM_getVolumeAtTag](#)
- interface [F_Tagged_AccumVM_SepVM_getVolumePtrAtIndex](#)
- interface [F_Tagged_AccumVM_SepVM_multiplyByVolume](#)
- interface [F_Tagged_AccumVM_SepVM_new](#)
- interface [F_Tagged_AccumVM_SepVM_normalizeByVolume](#)
- interface [getcentroidatindex](#)
- interface [getcentroidattag](#)
- interface [getcentroidptratindex](#)
- interface [getcobject](#)
- interface [getsize](#)
- interface [gettagforindex](#)
- interface [getvolumeatindex](#)
- interface [getvolumeattag](#)
- interface [getvolumeptratindex](#)
- interface [multiplybyvolume](#)
- interface [new](#)
- interface [normalizebyvolume](#)
- type [tagged_accumvm_sepvm_type](#)

Functions/Subroutines

- subroutine [tagged_accumvm_sepvm_class_new](#) (this)
- impure elemental subroutine [tagged_accumvm_sepvm_class_delete](#) (this)
- type([c_tagged_accumvm_sepvm](#)) function [tagged_accumvm_sepvm_class_getcobject](#) (this)
- subroutine [tagged_accumvm_sepvm_class_normalizebyvolume](#) (this)
- subroutine [tagged_accumvm_sepvm_class_multiplybyvolume](#) (this)
- real(irl_double) function [tagged_accumvm_sepvm_class_getvolumeatindex](#) (this, a_list_index, a_index)
- real(irl_double) function, dimension(3) [tagged_accumvm_sepvm_class_getcentroidatindex](#) (this, a_list← _index, a_index)
- real(irl_double) function [tagged_accumvm_sepvm_class_getvolumeattag](#) (this, a_tag, a_index)

- real(irl_double) function, dimension(3) **tagged_accumvm_sepvm_class_getcentroidattag** (this, a_tag, a_index)
- real(irl_double) function, pointer **tagged_accumvm_sepvm_class_getvolumepratindex** (this, a_list_index, a_index)
- real(irl_double) function, dimension(:), pointer **tagged_accumvm_sepvm_class_getcentroidptratindex** (this, a_list_index, a_index)
- integer(irl_unsignedindex_t) function **tagged_accumvm_sepvm_class_getsize** (this)
- integer(irl_unsignedindex_t) function **tagged_accumvm_sepvm_class_gettagforindex** (this, a_index)

5.32.1 Detailed Description

A fortran type class that allows the creation of IRL's AccumulatedVolumeMomentsM<SeparatedMoments<VolumeMoments>> class along with enabling some of its methods.

5.33 f_tagged_accumvm_vm_class Module Reference

A fortran type class that allows the creation of IRL's AccumulatedVolumeMomentsM<VolumeMoments> class along with enabling some of its methods.

Data Types

- type **c_tagged_accumvm_vm**
- interface **F_Tagged_AccumVM_VM_delete**
- interface **F_Tagged_AccumVM_VM_getCentroidAtIndex**
- interface **F_Tagged_AccumVM_VM_getCentroidPtrAtIndex**
- interface **F_Tagged_AccumVM_VM_getSize**
- interface **F_Tagged_AccumVM_VM_getTagForIndex**
- interface **F_Tagged_AccumVM_VM_getVolumeAtIndex**
- interface **F_Tagged_AccumVM_VM_getVolumePtrAtIndex**
- interface **F_Tagged_AccumVM_VM_multiplyByVolume**
- interface **F_Tagged_AccumVM_VM_new**
- interface **F_Tagged_AccumVM_VM_normalizeByVolume**
- interface **getcentroidatindex**
- interface **getcentroidptratindex**
- interface **getcobject**
- interface **getsize**
- interface **gettagforindex**
- interface **getvolumeatindex**
- interface **getvolumepratindex**
- interface **multiplybyvolume**
- interface **new**
- interface **normalizebyvolume**
- type **tagged_accumvm_vm_type**

Functions/Subroutines

- subroutine **tagged_accumvm_vm_class_new** (this)
- impure elemental subroutine **tagged_accumvm_vm_class_delete** (this)
- type([c_tagged_accumvm_vm](#)) function **tagged_accumvm_vm_class_getcobject** (this)
- subroutine **tagged_accumvm_vm_class_normalizebyvolume** (this)
- subroutine **tagged_accumvm_vm_class_multiplybyvolume** (this)
- real(irl_double) function **tagged_accumvm_vm_class_getvolumeatindex** (this, a_list_index)
- real(irl_double) function, dimension(3) **tagged_accumvm_vm_class_getcentroidatindex** (this, a_list_index)
- real(irl_double) function, pointer **tagged_accumvm_vm_class_getvolumepratindex** (this, a_list_index)
- real(irl_double) function, dimension(:,), pointer **tagged_accumvm_vm_class_getcentroidpratindex** (this, a_list_index)
- integer(irl_unsignedindex_t) function **tagged_accumvm_vm_class_getsize** (this)
- integer(irl_unsignedindex_t) function **tagged_accumvm_vm_class_gettagforindex** (this, a_index)

5.33.1 Detailed Description

A fortran type class that allows the creation of [IRL](#)'s AccumulatedVolumeMomentsM<VolumeMoments> class along with enabling some of its methods.

5.34 f_tet_class Module Reference

A fortran type class that allows the creation of [IRL](#)'s Tet class along with enabling some of its methods.

Data Types

- type [c_tet](#)
- interface [construct](#)
- interface [F_Tet_construct](#)
- interface [F_Tet_delete](#)
- interface [F_Tet_getBoundingPts](#)
- interface [F_Tet_new](#)
- interface [getboundingpts](#)
- interface [getcobject](#)
- interface [new](#)
- type [tet_type](#)

Functions/Subroutines

- subroutine **tet_class_new** (this)
- impure elemental subroutine **tet_class_delete** (this)
- type([c_tet](#)) function **tet_class_getcobject** (this)
- subroutine **tet_class_construct** (this, a_Tet_pts)
- subroutine **tet_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

5.34.1 Detailed Description

A fortran type class that allows the creation of [IRL](#)'s Tet class along with enabling some of its methods.

5.35 f_tri_class Module Reference

A fortran type class that allows the creation of IRL's Tri class along with enabling some of its methods.

Data Types

- type `c_tri`
- interface `calculateandsetplaneofexistence`
- interface `calculatecentroid`
- interface `calculatenormal`
- interface `calculatesign`
- interface `calculatevolume`
- interface `construct`
- interface `F_Tri_calculateAndSetPlaneOfExistence`
- interface `F_Tri_calculateCentroid`
- interface `F_Tri_calculateNormal`
- interface `F_Tri_calculateSign`
- interface `F_Tri_calculateVolume`
- interface `F_Tri_construct`
- interface `F_Tri_delete`
- interface `F_Tri_getBoundingPts`
- interface `F_Tri_getLocalizer`
- interface `F_Tri_getPlaneOfExistence`
- interface `F_Tri_getVertices`
- interface `F_Tri_new`
- interface `F_Tri_reversePtOrdering`
- interface `F_Tri_setPlaneOfExistence`
- interface `getboundingpts`
- interface `getcobject`
- interface `getlocalizer`
- interface `getplaneofexistence`
- interface `getvertices`
- interface `new`
- interface `reverseptordering`
- interface `setplaneofexistence`
- type `tri_type`

Functions/Subroutines

- subroutine `tri_class_new` (this)
- impure elemental subroutine `tri_class_delete` (this)
- type(`c_tri`) function `tri_class_getcobject` (this)
- subroutine `tri_class_construct` (this, a_pts)
- real(irl_double) function, dimension(1:3, 1:3) `tri_class_getvertices` (this)
- real(irl_double) function `tri_class_calculatevolume` (this)
- real(irl_double) function, dimension(1:3) `tri_class_calculatecentroid` (this)
- real(irl_double) function, dimension(1:3) `tri_class_calculatenormal` (this)
- subroutine `tri_class_getlocalizer` (this, a_planar_localizer)
- subroutine `tri_class_reverseptordering` (this)
- subroutine `tri_class_getboundingpts` (this, a_lower_pt, a_upper_pt)
- real(irl_double) function `tri_class_calculatesign` (this)
- subroutine `tri_class_setplaneofexistence` (this, a_plane)
- subroutine `tri_class_calculateandsetplaneofexistence` (this)
- real(irl_double) function, dimension(4) `tri_class_getplaneofexistence` (this)

5.35.1 Detailed Description

A fortran type class that allows the creation of IRL's Tri class along with enabling some of its methods.

5.36 f_vman_class Module Reference

A fortran type class that allows the creation of IRL's AccumulatedListedVolumeMomentsM<VolumeMomentsAndNormal> class along with enabling some of its methods.

Data Types

- type `c_vman`
- interface `F_VMAN_delete`
- interface `F_VMAN_getCentroid`
- interface `F_VMAN_getNormal`
- interface `F_VMAN_getVolume`
- interface `F_VMAN_multiplyByVolume`
- interface `F_VMAN_new`
- interface `F_VMAN_normalizeByVolume`
- interface `getcentroid`
- interface `getcobject`
- interface `getnormal`
- interface `getvolume`
- interface `multiplybyvolume`
- interface `new`
- interface `normalizebyvolume`
- type `vman_type`

Functions/Subroutines

- subroutine `vman_class_new` (this)
- impure elemental subroutine `vman_class_delete` (this)
- type(`c_vman`) function `vman_class_getcobject` (this)
- real(irl_double) function `vman_class_getvolume` (this)
- real(irl_double) function, dimension(3) `vman_class_getcentroid` (this)
- real(irl_double) function, dimension(3) `vman_class_getnormal` (this)
- subroutine `vman_class_normalizebyvolume` (this)
- subroutine `vman_class_multiplybyvolume` (this)

5.36.1 Detailed Description

A fortran type class that allows the creation of IRL's AccumulatedListedVolumeMomentsM<VolumeMomentsAndNormal> class along with enabling some of its methods.

5.37 f_volumefractionmatching Module Reference

This module contains mappings to the IRL C interface that deals with setting the distance to each plane in a reconstruction to recreate the volume fraction on the provided polyhedron.

Data Types

- interface [F_setDistanceToMatchVolumeFraction_RC_PS](#)
- interface [F_setDistanceToMatchVolumeFraction_RC_PS_DefTol](#)
- interface [setdistancetomatchvolumefraction](#)

Functions/Subroutines

- subroutine [setdistancetomatchvolumefraction_rc_ps](#) (a_rectangular_cuboid, a_volume_fraction, a_<planar_separator, a_volume_fraction_tolerance)
- subroutine [setdistancetomatchvolumefraction_rc_ps_deftol](#) (a_rectangular_cuboid, a_volume_fraction, a_planar_separator)

5.37.1 Detailed Description

This module contains mappings to the [IRL](#) C interface that deals with setting the distance to each plane in a reconstruction to recreate the volume fraction on the provided polyhedron.

5.38 IRL Namespace Reference

Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Classes

- class [AccumulatedListedVolumeMoments](#)
- class [AccumulatedVolumeMoments](#)
Self-expanding vector of VolumeMoments.
- struct [AccumulateIntoCollection](#)
- struct [AccumulateIntoScalar](#)
- class [AccumulateWrapper](#)

This is essentially a class that delays the evaluation of Tagged moments during generic_cutting with a ReconstructionLink for an eventual collection. As an example, this class needs to be used when cutting by a LocalizedSeparatorPath for tagged moments, since the first cutting operation (by the Localizer), is not what will dictate the tagging of the moments, but instead the second (by the PlanarSeparatorPath) does. We use of the tagged moments that are going to be returned need to be delayed and accumulated. We would then return them as auto moments = getVolumeMoments<AccumulateWrapper<TaggedAccumulatedVolumeMoments<VolumeMoments>>>(a_volume_to_cut, a_LocalizedSeparatorPath);.

- class [Accumulator](#)
This class is a SelfExpandingCollection that has operator += call the += operator for each member in its collection.
- class [AdvectedPlaneReconstruction](#)
- class [AdvectedPlaneReconstructionDebug](#)
- class [BFGS](#)
Implementation of BFGS method according to the book Numerical Optimization by Jorge Nocedal and Stephen Wright, 2006.
- class [Bisection](#)
A templated class driver that performs bisection.
- class [BlockObjectAllocation](#)
- class [BrentsMethod](#)

A templated class driver that performs Brent's Method.

- class [ByteBuffer](#)
- class [CappedDodecahedronSpecialization](#)
- class [CellCollection](#)

Class that contains a list of cells and coupled to moments.

- class [CellGroupedMoments](#)

A class that couples together a cell and another class.

- class [Centroid2D_Functor](#)
- class [Centroid3D_Functor](#)
- class [ChainedBlockStorage](#)
- class [Collection](#)

Essentially a std::vector with assertions on access through operator[] in debug mode.

- class [ConcaveBoxSpecialization](#)
- class [ConstIteratorThroughBracketOperator](#)
- class [ContinueDividingVolumeByPlanarReconstruction](#)
- struct [ContinueReducingVolumeToBeInternalToReconstruction](#)
- struct [CutSimplexByNextPlaneAccessedByIndex](#)
- class [CutTetVertices](#)
- class [CutTriangleVerticesAndPlane](#)
- class [DecomposedPolygonVertexStorage](#)
- class [DecomposedPolyhedronVertexStorage](#)
- class [DecomposedPolytopeVertexStorage](#)
- class [DelayedExpandableVertexAccess](#)
- class [DividedPolygonSpecialization](#)

A polygon that knows its centroid and can return the triangles that it is comprised of.

- class [DivideSimplexByPlanarReconstruction](#)
- class [DodecahedronSpecialization](#)
- struct [DoesNotHaveACollection](#)
- struct [DoesNotHaveALocalizer](#)
- struct [DoesNotHaveANestedType](#)
- struct [DoesNotHaveAReconstructionLink](#)
- struct [DoesNotHaveASeparator](#)
- struct [doNoEarlyBranching](#)
- struct [earlyBranchBothFullyBelowAndFullyAboveSimplicesLink](#)
- struct [earlyBranchBothFullyBelowAndFullyAboveSimplicesNoLink](#)
- class [ELVIRA_2D](#)
- class [ELVIRA_3D](#)
- class [ELVIRADebug](#)
- class [ELVIRADebug< ELVIRA_2D >](#)
- class [ELVIRADebug< ELVIRA_3D >](#)
- class [ELVIRANeighborhood](#)

Below plane volume fraction information in a stencil, to be used for ELVIRA. Up to 27 cells to cover 3x3x3 stencil in 3D. Can partially fill to 9 for 2D.

- class [ExpandableDividedPolygon](#)
- class [ExpandablePolygon](#)
- class [ExpandablePtList](#)
- struct [Expr](#)
- class [Face](#)
- class [GeneralPolygon](#)
- class [GeneralPolyhedron](#)
- class [HalfEdge](#)
- struct [HalfEdgeCutting](#)
- class [HalfEdgePolygon](#)
- class [HalfEdgePolyhedron](#)

- class [HalfEdgePolytope](#)
 - struct [has_localizer](#)
 - struct [has_localizer< const C >](#)
 - struct [has_localizer< LocalizedSeparator >](#)
 - struct [has_localizer< LocalizedSeparatorGroup >](#)
 - struct [has_localizer< LocalizedSeparatorGroupLink >](#)
 - struct [has_localizer< LocalizedSeparatorLink >](#)
 - struct [has_localizer< LocalizerLink >](#)
 - struct [has_localizer< LocalizerLinkFromLocalizedSeparatorLink >](#)
 - struct [has_localizer< MaskedLocalizedSeparatorLink >](#)
 - struct [has_localizer< PlanarLocalizer >](#)
 - struct [has_localizer< RestrictedLocalizerLinkFromLocalizedSeparatorLink >](#)
 - struct [has_localizer< RestrictedMaskedLocalizedSeparatorLink >](#)
 - struct [has_null_reconstruction](#)
 - struct [has_null_reconstruction< const C >](#)
 - struct [has_null_reconstruction< NullReconstruction >](#)
 - struct [has_separator](#)
 - struct [has_separator< const C >](#)
 - struct [has_separator< LocalizedSeparator >](#)
 - struct [has_separator< LocalizedSeparatorGroup >](#)
 - struct [has_separator< LocalizedSeparatorGroupLink >](#)
 - struct [has_separator< LocalizedSeparatorLink >](#)
 - struct [has_separator< MaskedLocalizedSeparatorLink >](#)
 - struct [has_separator< PlanarSeparator >](#)
 - struct [has_separator< PlanarSeparatorPath >](#)
 - struct [has_separator< RestrictedMaskedLocalizedSeparatorLink >](#)
 - struct [HasACollection](#)
 - struct [HasALocalizer](#)
 - struct [HasANestedType](#)
 - struct [HasAReconstructionLink](#)
 - struct [HasASeparator](#)
 - struct [HasLocalizer_AND_HasSeparator](#)
 - struct [HasLocalizer_OR_HasSeparator](#)
 - struct [HasLocalizer_XOR_HasSeparator](#)
 - class [HexahedronSpecialization](#)

A hexahedron class.
 - struct [IgnoreVolumeAbovePlane](#)
 - class [Illinois](#)

A templated class driver that performs the [Illinois](#) method of Regula-Falsi zero-finding.
 - struct [is_moments_collection](#)

Type trait to allow static checking that an object is a [AccumulatedVolumeMoments](#).
 - struct [is_moments_collection< AccumulatedListedVolumeMoments< VolumeMomentsType > >](#)

Any instantiation of [AccumulatedListedVolumeMoments](#) is a type of [volumeMomentsList](#).
 - struct [is_moments_collection< AccumulatedVolumeMoments< VolumeMomentsType > >](#)

Any instantiation of [AccumulatedVolumeMoments](#) is a type of [volumeMomentsList](#).
 - struct [is_moments_collection< const C >](#)
 - struct [is_moments_collection< ListedVolumeMoments< VolumeMomentsType > >](#)

Any instantiation of [ListedVolumeMoments](#) is a type of [volumeMomentsList](#).
 - struct [is_moments_collection< TaggedAccumulatedListedVolumeMoments< VolumeMomentsType > >](#)

Any instantiation of [AccumulatedListedVolumeMoments](#) is a type of [volumeMomentsList](#).
 - struct [is_moments_collection< TaggedAccumulatedVolumeMoments< VolumeMomentsType > >](#)
 - struct [is_nested_moments](#)
- Type trait to allow static checking that an object is a [AccumulatedVolumeMoments](#).*

- struct `is_nested_moments< AccumulateWrapper< NestedType > >`
Any instantiation of `AccumulatedVolumeMoments` is a type of `volumeMomentsList`.
- struct `is_nested_moments< const C >`
- struct `is_planar_separator`
- struct `is_planar_separator< const C >`
- struct `is_planar_separator< PlanarSeparator >`
- struct `is_polygon`
- struct `is_polygon< const C >`
- struct `is_polygon< ExpandableDividedPolygon< VertexType > >`
- struct `is_polygon< ExpandablePolygon< VertexType > >`
- struct `is_polygon< ProxyTri< GeometryType > >`
- struct `is_polygon< SegmentedDecomposedPolygon< VertexType > >`
- struct `is_polygon< SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices > >`
- struct `is_polygon< StoredTri< VertexType > >`
- struct `is_polyhedron`
- struct `is_polyhedron< const C >`
- struct `is_polyhedron< ProxyTet< GeometryType > >`
- struct `is_polyhedron< SegmentedDecomposedPolyhedron< VertexType > >`
- struct `is_polyhedron< SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices > >`
- struct `is_polyhedron< StoredCappedDodecahedron< VertexType > >`
- struct `is_polyhedron< StoredConcaveBox< VertexType > >`
- struct `is_polyhedron< StoredDodecahedron< VertexType > >`
- struct `is_polyhedron< StoredHexahedron< VertexType > >`
- struct `is_polyhedron< StoredPolyhedron24< VertexType > >`
- struct `is_polyhedron< StoredRectangularCuboid< VertexType > >`
- struct `is_polyhedron< StoredTet< VertexType > >`
- struct `is_reconstruction_link`
- struct `is_reconstruction_link< const C >`
- struct `is_reconstruction_link< LocalizedSeparatorGroupLink >`
- struct `is_reconstruction_link< LocalizedSeparatorLink >`
- struct `is_reconstruction_link< LocalizerLink >`
- struct `is_reconstruction_link< LocalizerLinkFromLocalizedSeparatorLink >`
- struct `is_reconstruction_link< MaskedLocalizedSeparatorLink >`
- struct `is_reconstruction_link< PlanarSeparatorPath >`
- struct `is_reconstruction_link< RestrictedLocalizerLinkFromLocalizedSeparatorLink >`
- struct `is_reconstruction_link< RestrictedMaskedLocalizedSeparatorLink >`
- struct `is_separated_moments`
Marking of if volume.
- struct `is_separated_moments< const C >`
- struct `is_separated_moments< SeparatedMoments< MomentsType > >`
- struct `is_tet`
- struct `is_tet< const C >`
- struct `is_tet< ProxyTet< GeometryType > >`
- struct `is_tet< StoredTet< VertexType > >`
- struct `is_tri`
- struct `is_tri< const C >`
- struct `is_tri< ProxyTri< GeometryType > >`
- struct `is_tri< StoredTri< VertexType > >`
- struct `isHalfEdgeCutting`
- struct `isHalfEdgeCutting< const C >`
- struct `isHalfEdgeCutting< HalfEdgeCutting >`
- struct `IsNotNullReconstruction`
- struct `IsNotAPlanarSeparator`
- struct `IsNullReconstruction`

- struct [IsPlanarSeparator](#)
- struct [IsRecursiveSimplexCutting](#)
- struct [IsRecursiveSimplexCutting< const C >](#)
- struct [IsRecursiveSimplexCutting< RecursiveSimplexCutting >](#)
- struct [IsSimplexCutting](#)
- struct [IsSimplexCutting< const C >](#)
- struct [IsSimplexCutting< SimplexCutting >](#)
- class [IterativeSolverForDistance](#)
Volume conserving distance-finding routine for two-plane reconstructions wrapped in a class.
- class [IteratorThroughBracketOperator](#)
- class [JoinedReconstructions](#)
Class that ties together two reconstructions. Will first cut by and use the current reconstruction, and then pass onwards to the next reconstruction.
- class [JoinedReconstructions< CurrentReconstruction, NullReconstruction >](#)
- class [JoinedReconstructionsToGroup](#)
- struct [KMeans](#)
A class that executes Kmeans when provided an appropriate DrivingClass.
- class [LevenbergMarquardt](#)
Levenberg-Marquardt optimization routine.
- class [LevenbergMarquardt< OptimizingClass, -1, kColumns >](#)
- class [LinearInterpolation_Functor](#)
- class [ListedVolumeMoments](#)
VolumeMoments wrapper for Lister class to add ability to normalize.
- class [Lister](#)
This class is a Collection that has operator += add the other Lister or object to the back of its own collection.
- class [LocalizerLinkFromLocalizedSeparatorLink](#)
- class [LVIRA_2D](#)
LVIRA class for reconstructions in 2D (x-y plane).
- class [LVIRA_3D](#)
LVIRA class for reconstructions in 3 dimensions.
- class [LVIRACommon](#)
Class to contain data and methods that will be used in all of the specific LVIRA optimization classes.
- class [LVIRADebug](#)
This class just calls the LVIRAType functions but allows debug statements to be printed. The solution path is also saved to be exported and visualized.
- class [LVIRANeighborhood](#)
Neighborhood storage used in the LVIRA optimization routines. This stores the CellGroupedMoments of the cell and the volume fraction.
- class [MaskedLocalizedSeparatorLink](#)
- class [MaskStripper](#)
- class [MathVector](#)
- class [MOF_2D](#)
- class [MOF_3D](#)
- class [MOFCommon](#)
- class [MOFDebug](#)
This class just calls the MOFType functions but allows debug statements to be printed.
- class [MomentCalculationType](#)
- class [MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polygon< ProxyType >::value > >](#)
- class [MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polyhedron< ProxyType >::value > >](#)
- class [MultiMaterialAccumulatedVolumeMoments](#)
This class is a specialization of TaggedAccumulatedVolumeMoments< MomentsType > that allows the computation of SeparatedMoments< MomentsType > for the target ID and the encompassing volume. Encompassing volume can be either given, or computed as the moments of all other phases.

- class [Normal](#)

A normal vector in 3D space.

- class [NullReconstruction](#)
- class [ObjectAllocationServer](#)
- class [PartitionByNormal](#)

This is a class that takes a list of objects and separates the list into two partitions, which are represented by the sums of the objects.

- struct [PassToNestedType](#)
- class [PathGraphNode](#)

Class to inherit from to provide linking between reconstructions, where the linking exists PER RECONSTRUCTION. This also follows a directed path-graph (linear graph), with a starting node and constant direction to the end node. The interface is kept the same as for the [UnDirectedGraphNode](#) in order to facilitate reuse in the cutting routines. All neighbor indices will simply point to the next node in the path-graph.

- class [PlanarLocalizer](#)

A planar representation of a convex polyhedron to localize integrations.

- class [PlanarReconstruction](#)
- class [PlanarSeparator](#)
- class [PlanarSeparatorPathGroup](#)

This is a class that helps organize PlanarSeparatorPath objects to be coherent in memory and facilitates their control more easily. It will continue to store all PlanarSeparatorPath objects that were added to it, but in most routine only the active ones will be used. Here, active means those that are present in priority_order_by_id_m.

- class [Plane](#)

Plane defined by $\mathbf{n} \cdot \mathbf{x} - d = 0$, where the normal points from liquid to gas.

- class [PolygonBase](#)
- class [PolygonConstructors](#)
- class [PolygonMomentsCalculation](#)
- class [PolygonMomentsCalculationCommon](#)
- class [PolygonSpecialization](#)

A polygon from a given plane.

- class [Polyhedron](#)
- class [Polyhedron24Specialization](#)
- class [PolyhedronMomentsCalculation](#)
- class [PolyhedronMomentsCalculation< Derived, PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, SimplexType >](#)
- class [PolyhedronMomentsCalculationCommon](#)
- class [ProxyPolygonConstructors](#)
- class [ProxyTet](#)

A tet from a collection of 4 points.

- class [ProxyTri](#)
- class [ProxyVertexAccess](#)
- class [Pt](#)

A point in 3D space.

- class [PtList](#)
- class [PtWithDataCommon](#)
- class [PtWithDoublesStatelessFunctor](#)
- class [R2P_2D1P](#)

R2P class for reconstructions in 2 dimensions with 1 plane (hence 2D1P)

- class [R2P_2D2P](#)

R2P class for reconstructions in 2 dimensions with 2 plane (hence 2D2P)

- class [R2P_3D1P](#)

R2P class for reconstructions in 3 dimensions with 1 plane (hence 3D1P)

- class [R2P_3D2P](#)

R2P class for reconstructions in 3 dimensions with 2 plane (hence 3D2P)

- class [R2PCommon](#)

Class to contain data and methods that will be used in all of the specific R2Poptimization.

- class [R2PDebug](#)

This class just calls the R2PType functions but allows debug statements to be printed.

- class [R2PNeighborhood](#)

Neighborhood storage used in the R2P optimization routines.

- class [ReconstructionLink](#)

This is a class template for linking together reconstructions that are already pointers to something else. This then does not require a different constructor because we are already taking pointers of the underlying reconstructions. For classes that are currently storing their own data, it would probably be best to generate a wrapping pointer class, which can be done with `JoinedReconstruction<ClassYouWant, NullReconstruction>`.

- class [RectangularCuboidBase](#)

- class [RectangularCuboidCommon](#)

- class [RectangularCuboidSpecialization](#)

A rectangular cuboid.

- struct [RecursiveSimplexCutting](#)

- class [ReferenceFrame](#)

A reference frame with three normals (1 for each direction in 3D space).

- class [ReferredToPlaneOFEistence](#)

- class [RestrictedLocalizerLinkFromLocalizedSeparatorLink](#)

- class [RestrictedMaskedLocalizedSeparatorLink](#)

- class [Secant](#)

A templated class driver that performs the `Secant` method for optimization.

- class [SegmentedDecomposedPolygon](#)

- class [SegmentedDecomposedPolyhedron](#)

- class [SegmentedDecomposedPolytope](#)

- class [SegmentedHalfEdgePolygon](#)

- class [SegmentedHalfEdgePolygonCommon](#)

- class [SegmentedHalfEdgePolygonSpecificPt](#)

- class [SegmentedHalfEdgePolygonSpecificPt< Pt, FaceType, VertexType, kMaxFaces, kMaxVertices >](#)

- class [SegmentedHalfEdgePolyhedron](#)

- class [SegmentedHalfEdgePolyhedronCommon](#)

- class [SegmentedHalfEdgePolyhedronSpecificPt](#)

- class [SegmentedHalfEdgePolyhedronSpecificPt< PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, FaceType, kMaxFaces, kMaxVertices >](#)

- class [SegmentedHalfEdgePolytope](#)

- class [SelfExpandingCollection](#)

A modified `Collection` class that self expands when using the operator[] would force out of memory access.

- class [SeparatedMoments](#)

- class [SeparatedMoments< VolumeMoments >](#)

- class [SeparatedMoments< VolumeMomentsAndDoubles< kArrayLength > >](#)

- class [SeparatedMomentsCommon](#)

Storage for multiple volume moments.

- struct [ShareVolumeAbovePlaneWithLinkedNeighbor](#)

- struct [SimplexCutting](#)

- struct [SimplexWrapper](#)

- struct [SimplexWrapper< SimplexType, enable_if_t< is_tet< SimplexType >::value > >](#)

- struct [SimplexWrapper< SimplexType, enable_if_t< is_tri< SimplexType >::value > >](#)

- class [SmallVector](#)

- class [StackVector](#)

- class [StolenGraph](#)

Class to inherit from that will utilize an already set up graph and mask some of its behaviors by shadowing in the `NodeType` class. In order to have a consistent interface with other graph classes, the `NodeType` class that inherits `StolenGraph` (which works due to CRTP) must implement a `NodeType getNeighbor(const UnsignedIndex_t a_neighbor_index) const` method See [planar_reconstruction/localizer_link_from_localized_separator_link.h](#) as an example.

- class [StoredCappedDodecahedron](#)
- class [StoredConcaveBox](#)
- class [StoredDodecahedron](#)
- class [StoredHexahedron](#)
- class [StoredPlaneOfExistence](#)
- class [StoredPolyhedron24](#)
- class [StoredRectangularCuboid](#)
- class [StoredTet](#)
- class [StoredTri](#)
- class [StoredVertexAccess](#)
- class [StrongWolfeConditionLineSearch](#)

Implementation of line search algorithm from "Line search algorithms with guaranteed sufficient decrease" by Jorge More and David Thuente, ACM Transactions on Mathematical Software, 1994. Some implementation details (such as the zoom function) are also taken from the book Numerical Optimization by Jorge Nocedal and Stephen Wright, 2006.

- class [TaggedAccumulatedListedVolumeMoments](#)
- class [TaggedAccumulatedVolumeMoments](#)
- class [TetSpecialization](#)
- class [TriSpecialization](#)
- class [UndirectedGraphNode](#)

Class to inherit from to provide linking between planes in a planar reconstruction to other reconstructions, where the linking exists PER PLANE.

- class [UnitQuaternion](#)

Unit quaternion to be used to perform rotations.
- class [vector_sum](#)
- class [Vertex](#)
- class [VertexList](#)
- class [VertexLocation](#)
- class [Volume](#)

A volume class which is just a double with special properties. Allows more general writing of functions.

- class [Volume2D_Functor](#)
- class [Volume3D_Functor](#)
- class [VolumeMoments](#)

Zeroeth (volume) and first order (centroid) geometric moments.

- class [VolumeMoments2D_Functor](#)
- class [VolumeMoments3D_Functor](#)
- class [VolumeMomentsAndDoubles](#)
- class [VolumeMomentsAndDoubles3D_Functor](#)
- class [VolumeMomentsAndNormal](#)

Class that contains [VolumeMoments](#) for a triangle and the triangle's normal.

- class [VolumeMomentsAndNormal2D_Functor](#)

Typedefs

- using **DefaultCuttingMethod** = [HalfEdgeCutting](#)
- using **ReconstructionDefaultCuttingMethod** = [RecursiveSimplexCutting](#)
- using **KeepOnlyInternalReconstructionVolumeWithNoEarlyBranch** = [ContinueDividingVolumeByPlanarReconstruction<CutSimplexByNextPlaneAccessedByIndex, doNoEarlyBranching, ContinueReducingVolumeToBeInternalToReconstruction, IgnoreVolumeAbovePlane >](#)
- using **SpreadVolumeThroughLinkedLocalizerNetworkWithNoEarlyBranch** = [ContinueDividingVolumeByPlanarReconstruction<CutSimplexByNextPlaneAccessedByIndex, doNoEarlyBranching, ContinueReducingVolumeToBeInternalToReconstruction, ShareVolumeAbovePlaneWithLinkedNeighbor >](#)

- using **KeepOnlyInternalReconstructionVolumeUsingEarlyBranch** = `ContinueDividingVolumeByPlanarReconstruction<CutSimplexByNextPlaneAccessedByIndex, earlyBranchBothFullyBelowAndFullyAboveSimplicesNoLink, ContinueReducingVolumeToBeInternalToReconstruction, IgnoreVolumeAbovePlane >`
- using **SpreadVolumeThroughLinkedLocalizerNetworkUsingEarlyBranch** = `ContinueDividingVolumeByPlanarReconstruction<CutSimplexByNextPlaneAccessedByIndex, earlyBranchBothFullyBelowAndFullyAboveSimplicesLink, ContinueReducingVolumeToBeInternalToReconstruction, ShareVolumeAbovePlaneWithLinkedNeighbor >`
- using **Vec3** = `MathVector< 3 >`
- using **DividedPolygon** = `ExpandableDividedPolygon< Pt >`
- using **Polygon** = `ExpandablePolygon< Pt >`
- template<class Derived , class VertexType >
using **ExpandableStorageAndStoredPlane** = `PolygonConstructors< Derived, VertexType, DelayedExpandableVertexAccess< Derived, VertexType, 0 >, StoredPlaneOfExistence >`
- template<class Derived , class VertexType , UnsignedIndex_t kMaskedVertices>
using **DelayedExpandableStorageAndStoredPlane** = `PolygonConstructors< Derived, VertexType, DelayedExpandableVertexAccess< Derived, VertexType, kMaskedVertices >, StoredPlaneOfExistence >`
- template<class Derived , class VertexType , UnsignedIndex_t kNumberOfVertices>
using **StaticStorageAndStoredPlane** = `PolygonConstructors< Derived, VertexType, StoredVertexAccess< Derived, VertexType, kNumberOfVertices >, StoredPlaneOfExistence >`
- template<class Derived , class GeometryType , UnsignedIndex_t kNumberOfVertices>
using **ProxyStaticStorageAndReferredToPlane** = `ProxyPolygonConstructors< Derived, GeometryType, kNumberOfVertices, ReferredToPlaneOfExistence >`
- using **Tri** = `StoredTri< Pt >`
- using **CappedDodecahedron** = `StoredCappedDodecahedron< Pt >`
- using **ConcaveBox** = `StoredConcaveBox< Pt >`
- using **Dodecahedron** = `StoredDodecahedron< Pt >`
- using **Hexahedron** = `StoredHexahedron< Pt >`
- using **Polyhedron24** = `StoredPolyhedron24< Pt >`
- using **RectangularCuboid** = `StoredRectangularCuboid< Pt >`
- using **Tet** = `StoredTet< Pt >`
- template<bool Cond, typename T = void>
using **enable_if_t** = `typename std::enable_if< Cond, T >::type`
- using **UnsignedIndex_t** = `uint32_t`
- using **LargeOffsetIndex_t** = `std::size_t`
- using **LookupIndex_t** = `uint8_t`
- using **Byte_t** = `unsigned char`
- using **LocalizedSeparator** = `JoinedReconstructions< PlanarLocalizer, PlanarSeparator >`
- using **LocalizedSeparatorGroup** = `JoinedReconstructionsToGroup< PlanarLocalizer, PlanarSeparatorPathGroup >`
- using **LocalizedSeparatorGroupLink** = `ReconstructionLink< LocalizedSeparatorGroup, UnDirectedGraphNode >`
- using **LocalizedSeparatorLink** = `ReconstructionLink< LocalizedSeparator, UnDirectedGraphNode >`
- using **LocalizerLink** = `ReconstructionLink< JoinedReconstructions< PlanarLocalizer, NullReconstruction >, UnDirectedGraphNode >`
- using **PlanarSeparatorPath** = `ReconstructionLink< JoinedReconstructions< PlanarSeparator, NullReconstruction >, PathGraphNode >`

This class handles the ability to chain together `PlanarSeparator` objects. Currently, due to restrictions in the implementation of the cutting routines, these `PlanarSeparator` objects are limited to a single plane. In actuality, the limitation is on requiring that the planes represent a convex volume (meaning `flip_cut_m = 1.0`). This is not a fundamental limitation, but simply requires some additional work in the cutting routines to enable the use of `PlanarSeparators` representing non-convex volumes. The necessity to have a single plane is enforced through assertions. Make sure to not run without assertions off and multiple-plane `PlanarSeparators`!

Enumerations

- enum **c_RuntimeCuttingMethod** { `RecursiveSimplexCutting` = 0, `HalfEdgeCutting` = 1, `SimplexCutting` = 2 }

Functions

- template<class ReturnType , class EncompassingType , class ReconstructionType >
static ReturnType **c_RuntimegetNormalizedVolumeMoments** (const EncompassingType &a_
encompassing_polytope, const ReconstructionType &a_reconstruction, const c_RuntimeCuttingMethod &a_cutting_method)
- template<class ReturnType , class EncompassingType , class ReconstructionType >
static ReturnType **c_RuntimegetVolumeMoments** (const EncompassingType &a_encompassing_polytope, const ReconstructionType &a_reconstruction, const c_RuntimeCuttingMethod &a_cutting_method)
- template<class CuttingType >
constexpr c_RuntimeCuttingMethod **c_getCompiledDefaultCuttingMethod** (void)
- template<>
constexpr c_RuntimeCuttingMethod **c_getCompiledDefaultCuttingMethod**< **RecursiveSimplexCutting** > (void)
- template<>
constexpr c_RuntimeCuttingMethod **c_getCompiledDefaultCuttingMethod**< **HalfEdgeCutting** > (void)
- template<>
constexpr c_RuntimeCuttingMethod **c_getCompiledDefaultCuttingMethod**< **SimplexCutting** > (void)
- template<class ContainerType >
IteratorThroughBracketOperator< ContainerType > **operator+** (const **IteratorThroughBracketOperator**< ContainerType > &a_iterator, const std::ptrdiff_t a_increment)
- template<class ContainerType >
IteratorThroughBracketOperator< ContainerType > **operator+** (const std::ptrdiff_t a_increment, const **IteratorThroughBracketOperator**< ContainerType > &a_iterator)
- template<class ContainerType >
IteratorThroughBracketOperator< ContainerType > **operator-** (const **IteratorThroughBracketOperator**< ContainerType > &a_iterator, const std::ptrdiff_t a_increment)
- template<class ContainerType >
IteratorThroughBracketOperator< ContainerType > **operator-** (const std::ptrdiff_t a_increment, const **IteratorThroughBracketOperator**< ContainerType > &a_iterator)
- template<class ContainerType >
ConstIteratorThroughBracketOperator< ContainerType > **operator+** (const **ConstIteratorThroughBracketOperator**< ContainerType > &a_iterator, const std::ptrdiff_t a_increment)
- template<class ContainerType >
ConstIteratorThroughBracketOperator< ContainerType > **operator+** (const std::ptrdiff_t a_increment, const **ConstIteratorThroughBracketOperator**< ContainerType > &a_iterator)
- template<class ContainerType >
ConstIteratorThroughBracketOperator< ContainerType > **operator-** (const **ConstIteratorThroughBracketOperator**< ContainerType > &a_iterator, const std::ptrdiff_t a_increment)
- template<class ContainerType >
ConstIteratorThroughBracketOperator< ContainerType > **operator-** (const std::ptrdiff_t a_increment, const **ConstIteratorThroughBracketOperator**< ContainerType > &a_iterator)
- template<class ContainerType >
std::ptrdiff_t **operator-** (const **ConstIteratorThroughBracketOperator**< ContainerType > &a_iterator_0, const **ConstIteratorThroughBracketOperator**< ContainerType > &a_iterator_1)
- template<class ObjectType , UnsignedIndex_t kMaxSize>
std::ostream & **operator<<** (std::ostream &out, const **StackVector**< ObjectType, kMaxSize > &a_stack_
vector)
- template<class PolygonType , class ReconstructionType >
PolygonType cutPolygonByReconstruction (const PolygonType &a_polygon, const **Plane** *a_plane_that_
_created_polygon, const ReconstructionType &a_reconstruction_to_intersect)
- template<class PolygonType , class ReconstructionType >
void cutPolygonByReconstructionInPlace (PolygonType *a_polygon, const **Plane** *a_plane_that_
_created_polygon, const ReconstructionType &a_reconstruction_to_intersect)
- template<class PolygonType , class HexahedronType >
PolygonType cutPlaneByHexahedron (const HexahedronType &a_hexahedron, const **Plane** &a_plane)

Cuts a plane by a_hexahedron, returning a PlanePolygon which is a polygon of the plane contained inside the rectangular_cuboid.

- template<class PolygonType >
PolygonType **cutPlaneByPolyhedron** (const [RectangularCuboid](#) &a_bouding_box, const [Plane](#) &a_plane)
- template<class PolygonType >
PolygonType **cutPlaneByPolyhedron** (const [Hexahedron](#) &a_bouding_box, const [Plane](#) &a_plane)
- template<class PolygonType , class HexahedronType >
PolygonType **cutPlaneByPolyhedron** (const HexahedronType &a_bouding_box, const [PlanarLocalizer](#) &a_planar_localizer, const [Plane](#) &a_plane)
- template<class PolygonType >
PolygonType **cutPlaneByPolyhedron** (const [RectangularCuboid](#) &a_bouding_box, const [PlanarLocalizer](#) &a_planar_localizer, const [Plane](#) &a_plane)
- template<class PolygonType >
PolygonType **cutPlaneByPolyhedron** (const [Hexahedron](#) &a_bouding_box, const [PlanarLocalizer](#) &a_planar_localizer, const [Plane](#) &a_plane)
- template<class PolygonType >
void **signedDistanceToPolygonVertices** (const [PolygonType](#) &a_polygon, const [Plane](#) &a_cutting_plane, const double a_flip_cut, std::array<double, 6> *a_distance_to_vertices)

Loops through the vertices of a polygon and computes a signed distance from a_cutting_plane for each. Also sets all remaining distances in array to distance value of first vertex to be cyclic.

- template<class PolygonType >
PolygonType **cutPolygonByPlane** (const [PolygonType](#) &a_polygon, const [Plane](#) &a_cutting_plane, const double a_flip_cut)

Cuts a PlanePolygon by a given plane, returning a new Polygon of type PolygonType which is a polygon of the plane after being intersected with a_cutting_plane.

- template<class PolygonType >
void **cutPolygonByPlaneInPlace** (const [PolygonType](#) *a_polygon, const [Plane](#) &a_cutting_plane, const double a_flip_cut)
- template<class PolygonType , class PolyhedronType >
PolygonType **getPlanePolygonFromReconstruction** (const PolyhedronType &a_polyhedron, const [PlanarSeparator](#) &a_reconstruction, const [Plane](#) &a_plane_to_make_polygon)
- template<class PolygonType >
Return ConvexPolygon that represents the plane in a_reconstruction[a_plane_to_make_polygon] on `a_hexahedron.
- template<class PolygonType >
PolygonType **getPlanePolygonFromReconstruction** (const [RectangularCuboid](#) &a_polyhedron, const [PlanarSeparator](#) &a_reconstruction, const [Plane](#) &a_plane_to_make_polygon)
- template<class PolygonType >
PolygonType **getPlanePolygonFromReconstruction** (const [Hexahedron](#) &a_polyhedron, const [PlanarSeparator](#) &a_reconstruction, const [Plane](#) &a_plane_to_make_polygon)
- template<class PolygonType , class PolyhedronType >
PolygonType **getPlanePolygonFromReconstruction** (const PolyhedronType &a_polyhedron, const [PlanarLocalizer](#) &a_polyhedron_localizer, const [PlanarSeparator](#) &a_reconstruction, const [Plane](#) &a_plane_to_make_polygon)
- template<class PolygonType >
PolygonType **getPlanePolygonFromReconstruction** (const [RectangularCuboid](#) &a_polyhedron, const [PlanarLocalizer](#) &a_polyhedron_localizer, const [PlanarSeparator](#) &a_reconstruction, const [Plane](#) &a_plane_to_make_polygon)
- template<class PolygonType >
PolygonType **getPlanePolygonFromReconstruction** (const [Hexahedron](#) &a_polyhedron, const [PlanarLocalizer](#) &a_polyhedron_localizer, const [PlanarSeparator](#) &a_reconstruction, const [Plane](#) &a_plane_to_make_polygon)
- template<class HexahedronType >
double **getReconstructionSurfaceArea** (const HexahedronType &a_hexahedron, const [PlanarSeparator](#) &a_reconstruction)

Cut a supplied rectangular cuboid by a given reconstruction and return the surface area.

- template<class ReturnType , class CuttingMethod , class EncompassingType >
`__attribute__ ((pure)) __attribute__((hot)) inline ReturnType getVolumeMoments(const EncompassingType &a_encompassing_polyhedron)`
- template<class ReturnType , class CuttingMethod , class SegmentedPolytopeType , class HalfEdgePolytopeType >
`__attribute__ ((hot)) inline ReturnType getVolumeMoments(SegmentedPolytopeType *a_polytope)`
- template<class ReturnType , class EncompassingType >
`__attribute__ ((pure)) __attribute__((hot)) inline ReturnType getVolumeMoments(const EncompassingType &a_encompassing_polyhedron)`
- template<class ReturnType , class CuttingMethod = DefaultCuttingMethod, class EncompassingType , class ReconstructionType >
`__attribute__ ((pure)) __attribute__((hot)) inline ReturnType getVolumeMoments(const EncompassingType &a_encompassing_polyhedron)`
- template<class ReturnType , class CuttingMethod = DefaultCuttingMethod, class SegmentedPolytopeType , class HalfEdgePolytopeType , class ReconstructionType >
`__attribute__ ((hot)) inline ReturnType getVolumeMoments(SegmentedPolytopeType *a_polytope)`
- template<class SegmentedHalfEdgePolyhedronType , class HalfEdgePolytopeType >
`enable_if_t< is_polyhedron< SegmentedHalfEdgePolyhedronType >::value > splitHalfEdgePolytope (SegmentedHalfEdgePolyhedronType *a_polytope, SegmentedHalfEdgePolyhedronType *a_clipped_polytope, HalfEdgePolytopeType *a_complete_polytope, const Plane &a_plane)`
- template<class SegmentedHalfEdgePolygonType , class HalfEdgePolytopeType >
`enable_if_t< is_polygon< SegmentedHalfEdgePolygonType >::value > splitHalfEdgePolytope (SegmentedHalfEdgePolygonType *a_polytope, SegmentedHalfEdgePolygonType *a_clipped_polytope, HalfEdgePolytopeType *a_complete_polytope, const Plane &a_plane)`
- template<class SegmentedHalfEdgePolyhedronType , class HalfEdgePolytopeType >
`enable_if_t< is_polyhedron< SegmentedHalfEdgePolyhedronType >::value > truncateHalfEdgePolytope (SegmentedHalfEdgePolyhedronType *a_polytope, HalfEdgePolytopeType *a_complete_polytope, const Plane &a_plane)`
- template<class SegmentedHalfEdgePolygonType , class HalfEdgePolytopeType >
`enable_if_t< is_polygon< SegmentedHalfEdgePolygonType >::value > truncateHalfEdgePolytope (SegmentedHalfEdgePolygonType *a_polytope, HalfEdgePolytopeType *a_complete_polytope, const Plane &a_plane)`
- template<class SegmentedPolytopeType , class HalfEdgePolytopeType , class ReconstructionType , class ReturnType >
`void getVolumeMomentsForPolytope (SegmentedPolytopeType *a_polytope, HalfEdgePolytopeType *a_complete_polytope, const ReconstructionType &a_reconstruction, ReturnType *a_moments_to_return)`
- template<class HalfEdgeType , class SegmentedHalfEdgePolytopeType , class HalfEdgePolytopeType >
`void subdivideEdge (HalfEdgeType *a_half_edge_with_intersection, SegmentedHalfEdgePolytopeType *a_polytope, HalfEdgePolytopeType *a_complete_polytope)`
- template<class HalfEdgeType , class SegmentedHalfEdgePolygonType , class HalfEdgePolytopeType >
`HalfEdgeType * separateIntersectedHalfEdge (HalfEdgeType *a_half_edge_with_intersection, SegmentedHalfEdgePolygonType *a_polytope, HalfEdgePolytopeType *a_complete_polytope)`
- template<class HalfEdgeType , class HalfEdgePolytopeType >
`HalfEdgeType * createOppositeHalfEdgeFromIntersection (HalfEdgeType *a_half_edge_with_intersection, HalfEdgeType *a_newly_created_half_edge, HalfEdgePolytopeType *a_complete_polytope)`
- template<class HalfEdgeType >
`void correctConnectivityAfterIntersection (HalfEdgeType *a_half_edge_with_intersection, HalfEdgeType *a_newly_created_half_edge, HalfEdgeType *a_half_edge_with_intersection_opposite, HalfEdgeType *opposite_half_edge_intersection)`
- template<class ReturnType , class EncompassingType , class ReconstructionType >
`ReturnType cutThroughHalfEdgeStructures (const EncompassingType &a_polytope, const ReconstructionType &a_reconstruction)`
- template<class SimplexType , class ReconstructionType , class ReturnType >
`void getVolumeMomentsForSimplex (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, ReturnType *a_moments_to_return)`
- template<class ReturnType , class EncompassingType , class ReconstructionType >
`ReturnType cutThroughRecursiveSimplex (const EncompassingType &a_encompassing_polyhedron, const ReconstructionType &a_separating_reconstruction)`
- template<class SegmentedHalfEdgePolyhedronType , class HalfEdgePolytopeType >
`enable_if_t< is_polyhedron< SegmentedHalfEdgePolyhedronType >::value > splitDecomposedPolytope`

- (SegmentedHalfEdgePolyhedronType *a_polytope, SegmentedHalfEdgePolyhedronType *a_clipped_<polytope, HalfEdgePolytopeType *a_complete_polytope, const Plane &a_plane)
- template<class SegmentedHalfEdgePolygonType , class HalfEdgePolytopeType >
 enable_if_t< **is_polygon**< SegmentedHalfEdgePolygonType >::value > **splitDecomposedPolytope** (SegmentedHalfEdgePolygonType *a_polytope, SegmentedHalfEdgePolygonType *a_clipped_polytope, HalfEdgePolytopeType *a_complete_polytope, const Plane &a_plane)
 - template<class SegmentedHalfEdgePolyhedronType , class HalfEdgePolytopeType >
 enable_if_t< **is_polyhedron**< SegmentedHalfEdgePolyhedronType >::value > **truncateDecomposedPolytope** (SegmentedHalfEdgePolyhedronType *a_polytope, HalfEdgePolytopeType *a_complete_polytope, const Plane &a_plane)
 - template<class SegmentedHalfEdgePolygonType , class HalfEdgePolytopeType >
 enable_if_t< **is_polygon**< SegmentedHalfEdgePolygonType >::value > **truncateDecomposedPolytope** (SegmentedHalfEdgePolygonType *a_polytope, HalfEdgePolytopeType *a_complete_polytope, const Plane &a_plane)
 - template<class SegmentedPolytopeType , class HalfEdgePolytopeType , class ReconstructionType , class ReturnType >
 void **getVolumeMomentsForDecomposedPolytope** (SegmentedPolytopeType *a_polytope, HalfEdgePolytopeType *a_complete_polytope, const ReconstructionType &a_reconstruction, ReturnType *a_moments_to_return)
 - template<class ReturnType , class EncompassingType , class ReconstructionType >
 ReturnType **cutThroughSimplex** (const EncompassingType &a_encompassing_polyhedron, const ReconstructionType &a_separating_reconstruction)
 - template<UnsignedIndex_t kNumberOfElements>
 attribute ((const)) inline double operator*(const MathVector<kNumberOfElements > &a_vec_0
 *Overload * operator to be dot product of the two vectors.*)
 - template<UnsignedIndex_t kNumberOfElements>
 attribute ((const)) inline MathVector<kNumberOfElements > operator*(const double a_double
 *Overload * operator between double and MathVector to return a MathVector.*)
 - template<class GeometryType , class CalculationFunctor >
 auto **calculateMoments** (const GeometryType &a_geometry, CalculationFunctor a_moment_accumulator)
 -> typename CalculationFunctor::ReturnType
 - double **operator*** (const Normal &a_normal_0, const Normal &a_normal_1)
 *Overload * operator to be dot product of the two normal vectors.*
 - double **operator*** (const Pt &a_pt, const Normal &a_normal)
 *Overload * operator between Pt and Normal to be a dot product.*
 - double **operator*** (const Normal &a_normal, const Pt &a_pt)
 *Overload * operator between Pt and Normal to be a dot product.*
 - Normal **operator*** (const double a_double, const Normal &a_normal)
 *Overload * operator between double and Normal to return a Normal.*
 - Normal **operator*** (const Normal &a_normal, const double a_double)
 *Overload * operator between double and Normal to be return a Normal.*
 - Normal **operator/** (const Normal &a_normal, const double a_double)
 *Overload * operator between double and Normal to be return a Normal.*
 - std::ostream & **operator<<** (std::ostream &out, const Normal &a_normal)
 For printing out a normal to a stream.
 - std::ostream & **operator<<** (std::ostream &out, const Plane &a_plane)
 - std::ostream & **operator<<** (std::ostream &out, const Pt &a_pt)
 - Pt **operator-** (const Pt &a_pt1, const Pt &a_pt2)
 overloaded - operator that performs element-wise subtraction.
 - Pt **operator*** (const double &a_multiplier, const Pt &a_pt2)
 *overloaded * operator that multiplies each element in loc_m by a_multiplier.*
 - Pt **operator*** (const Pt &a_pt, const double &a_multiplier)
 *overloaded * operator that multiplies each element in loc_m by a_multiplier.*
 - template<class FunctorType , UnsignedIndex_t kArrayLength>
 std::ostream & **operator<<** (std::ostream &out, const PtWithDoublesStatelessFunctor< FunctorType, k->ArrayLength > &a_pt_with_data)

- `UnitQuaternion rotateNormalOntoNormal (const Normal &a_normal_original, const Normal &a_normal_destination)`
Same as other function, however does not return the angle of rotation or rotation axis.
- `UnitQuaternion rotateNormalOntoNormal (const Normal &a_normal_original, const Normal &a_normal_destination, double *a_rotation_amount, Normal *a_rotation_axis)`
Return the quaternion that would rotate a_normal_original onto a_normal_destination.
- `ReferenceFrame getOrthonormalSystem (const Normal &a_normal)`
Create an orthonormal reference frame with axis 3 being a_normal.
- `Normal getSharedNormal (const Normal &a_normal_0, const Normal &a_normal_1)`
Same as other function, however does not return the half-angle of rotation or rotation axis.
- `Normal getSharedNormal (const Normal &a_normal_0, const Normal &a_normal_1, double *a_half_rotation_angle, Normal *a_rotation_axis)`
Get normal that is halfway between a_normal_0 and a_normal_1. Return through pointer the rotation to the shared normal and the axis of rotation.
- template<class FaceType>
`FaceType & getOpenBoundaryFace (void)`
- template<class VertexType>
`void doubleLinkHalfEdges (HalfEdge< VertexType > *a_starting_half_edge, HalfEdge< VertexType > *a_ending_half_edge)`
- template<class PtType, class VertexLocationType, class VertexType, class HalfEdgeType, class FaceType, UnsignedIndex_t kMaxHalfEdges, UnsignedIndex_t kMaxVertices, UnsignedIndex_t kMaxFaces>
`std::ostream & operator<< (std::ostream &out, const HalfEdgePolytope< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces > &a_polyhedron)`
- template<class FaceType, class VertexType, UnsignedIndex_t kMaxFaces, UnsignedIndex_t kMaxVertices>
`std::ostream & operator<< (std::ostream &out, const SegmentedHalfEdgePolytope< FaceType, VertexType, kMaxFaces, kMaxVertices > &a_polyhedron)`
- template<class Derived, class VertexType, class SimplexType>
`std::ostream & operator<< (std::ostream &out, const GeneralPolygon< Derived, VertexType, SimplexType > &a_polygon_base)`
- template<class Derived, class VertexType, class SimplexType>
`std::ostream & operator<< (std::ostream &out, const GeneralPolyhedron< Derived, VertexType, SimplexType > &a_polyhedron)`
- `double distanceBetweenPts (const Pt &a_pt_1, const Pt &a_pt_2)`
Calculates the distance between two points.
- `double squaredDistanceBetweenPts (const Pt &a_pt_1, const Pt &a_pt_2)`
Calculates the squared distance between two points.
- `void makeNormalFaceOutwardsFromVolume (const Pt &a_volume_centroid, const Pt &a_surface_centroid, Normal *a_normal)`
Correct normal to point outwards from the object, i.e. in the same general direction as a_surface_centroid - a_volume_centroid.
- template<class DistanceListType>
`LookupIndex_t getGeometricCasId (const DistanceListType &a_distances)`
*Given an array of distances to an intersecting plane and the length of that array, a unique integer ID is calculated based on the sign and index of each element in a_distances. This is often used to then reference unique cases in a lookup table. The ID is calculated as $\sum_{i=0}^{i=a_number_of_distances} 2^i * (0.5 + \text{sign}(0.5, a_distances[i]))$.*
- template<class GeometryType>
`bool isPlaneIntersectingCell (const Plane &a_plane, const GeometryType &a_geometry)`
Checks to see if a plane intersects a given cell.
- template<class ReconstructionType>
`bool isPtInternal (const Pt &a_pt, const ReconstructionType &a_reconstruction)`
Checks to see if point is internal to PlanarReconstruction.
- template<class GeometryType, class VertexDistanceList>
`void signedDistanceToVertices (const GeometryType &a_geometry, const Plane &a_cutting_planem, VertexDistanceList *a_distance_to_vertices)`

- Loops through the vertices of the supplied geometry and computes a signed distance from a_cutting_plane for each.*
- template<class EncompassingType , class ReconstructionType >
int **checkPureObject** (const EncompassingType &a_encompassing_polyhedron, const ReconstructionType &a_reconstruction)
Test if the object passed to it lays entirely on one side of the reconstruction.
 - double **safelyTiny** (const double a_value)
Takes max between abs(a_value) and DBL_MIN while preserving sign.
 - double **safelyEpsilon** (const double a_value)
Takes max between abs(a_value) and DBL_EPSILON while preserving sign.
 - double **safelySmall** (const double a_value, const double a_small_value)
Takes max between abs(a_value) and the_smallest_value while preserving sign.
 - double **clipBetween** (const double a_smallest_value, const double a_value, const double a_largest_value)
Clip value to lay between the min and max values.
 - bool **wantPurelyInternal** (const double a_internal_fraction)
Returns whether the liquid volume fraction indicates full liquid.
 - bool **wantPurelyExternal** (const double a_internal_fraction)
Returns whether the liquid volume fraction indicates full gas.
 - void **sort3Ascending** (double *a_items)
Sort 3 doubles in an array into ascending order via insertion.
 - void **sort3Descending** (double *a_items)
Sort 3 doubles in an array into descending order via insertion.
 - void **sort3AscendingTracked** (double *a_items, int *a_original_index)
Sort 3 doubles in an array into ascending order via insertion. The original location of the item is also tracked to allow reversing.
 - void **sort3DescendingTracked** (double *a_items, int *a_original_index)
Sort 3 doubles in an array into descending order via insertion. The original location of the item is also tracked to allow reversing.
 - template<class CarriedType , class DictatingType >
void **sortAscendingBasedOnOtherArray** (CarriedType *a_carried_array, DictatingType *a_dictating_array)
This function sorts a_carried_array based on the elements in a_dictating_array using insertion sort.
 - template<class CarriedType , class DictatingType >
void **sortDescendingBasedOnOtherArray** (CarriedType *a_carried_array, DictatingType *a_dictating_array)
This function sorts a_carried_array based on the elements in a_dictating_array using insertion sort.
 - template<typename V1 , typename V2 >
enable_if_t< std::is_base_of< Expr< V1 >, V1 >::value && std::is_base_of< Expr< V2 >, V2 >::value,
vector_sum< V1, V2 > > **operator+** (const Expr< V1 > &x, const Expr< V2 > &y)
 - constexpr double **deg2Rad** (const double a_degree)
Convert degrees to radians.
 - constexpr double **rad2Deg** (const double a_radian)
Convert radians to degrees.
 - double **angleNormalize** (const double a_radian)
Normalize a given angle (in radians) to be between 0 and 2 pi.
 - double **signedAngleNormalize** (const double a_radian)
Normalize a given angle (in radians) to be between -2 pi and 2 pi.
 - template<class DataType >
double **magnitude** (const DataType &a_vector_0, const DataType &a_vector_1)
Calculate the magnitude between two 3 element vectors.
 - template<class DataType >
double **squaredMagnitude** (const DataType &a_vector)
Calculate the squared magnitude between for a 3 element vector.
 - template<class DataType >
DataType **crossProduct** (const DataType &a_vector_0, const DataType &a_vector_1)

- *Take cross product of two 3-element vectors.*
- template<class **DataType**>
DataType crossProductNormalized (const **DataType** &a_vector_0, const **DataType** &a_vector_1)
Cross product of two 3-element vectors then normalized.
- template<class **DataType**>
double dotProduct (const **DataType** &a_vector_0, const **DataType** &a_vector_1)
Dot product between two 3 element vectors.
- template<class **DataType**>
double scalarTripleProduct (const **DataType** &a_vector_0, const **DataType** &a_vector_1, const **DataType** &a_vector_2)
Scalar triple product of 3, 3 element vectors.
- template<class **ObjectType**, class **ContainerType**>
void serializeAndPack (const **ObjectType** &a_object, **ContainerType** *a_container)
- template<class **ObjectType**, class **ContainerType**>
void unpackAndStore (**ObjectType** *a_object, **ContainerType** *a_container)
- **double findDistanceOnePlane** (const **RectangularCuboid** &a_rectangular_cuboid, const **double** a_volume_fraction, const **Normal** &a_normal)
Volume conserving distance-finding routine for single-plane reconstructions.
- **double getAlpha** (const **double** *a_mm, const **double** a_VOFo)
Get distance in local coordinate system according to Scardovelli & Zaleski, JCP 164,228-247 (2000)
- template<class **CellType**, class **ReconstructionType**>
void runIterativeSolverForDistance (const **CellType** &a_cell, const **double** a_volume_fraction, **ReconstructionType** *a_reconstruction, const **double** a_volume_fraction_tolerance=global_constants::TWO_PLANE_DISTANCE_VOLUME_FRACTION_TOLERANCE)
- template<class **CellType**, class **ReconstructionType**>
void cleanReconstruction (const **CellType** &a_cell, const **double** a_volume_fraction, **ReconstructionType** *a_reconstruction)
- template<class **CellType**, class **ReconstructionType**>
void cleanReconstructionOutOfCell (const **CellType** &a_cell, const **double** a_volume_fraction, **ReconstructionType** *a_reconstruction)
- template<class **CellType**, class **ReconstructionType**>
void cleanReconstructionSameNormal (const **CellType** &a_cell, const **double** a_volume_fraction, **ReconstructionType** *a_reconstruction)
- template<class **ReconstructionType**>
void setToPurePhaseReconstruction (const **double** a_internal_volume_fraction, **ReconstructionType** *a_reconstruction)
- template<class **CellType**>
PlanarSeparator reconstructionWithR2P2D (const **R2PNeighborhood**< **CellType** > &a_neighborhood_geometry, **PlanarSeparator** a_initial_reconstruction)
Perform R2P reconstruction for a 2D problem in the x-y plane.
- template<class **CellType**>
PlanarSeparator reconstructionWithR2P3D (const **R2PNeighborhood**< **CellType** > &a_neighborhood_geometry, **PlanarSeparator** a_initial_reconstruction)
Perform R2P reconstruction for a 3D problem.
- **PlanarSeparator reconstructionWithELVIRA2D** (const **ELVIRANeighborhood** &a_neighborhood_geometry)
Perform ELVIRA Reconstruction for 2D.
- **PlanarSeparator reconstructionWithELVIRA3D** (const **ELVIRANeighborhood** &a_neighborhood_geometry)
Perform ELVIRA Reconstruction for 3D.
- template<class **CellType**>
PlanarSeparator reconstructionWithLVIRA2D (const **LVIRANeighborhood**< **CellType** > &a_neighborhood_geometry, **PlanarSeparator** a_initial_reconstruction)
Perform LVIRA Reconstruction for 2D.
- template<class **CellType**>
PlanarSeparator reconstructionWithLVIRA3D (const **LVIRANeighborhood**< **CellType** > &a_neighborhood_geometry, **PlanarSeparator** a_initial_reconstruction)

- template<class CellType >
`PlanarSeparator reconstructionWithMOF2D (const CellType &a_cell, const SeparatedMoments< VolumeMoments > &a_svm, const double a_internal_weight=0.5, const double a_external_weight=0.5)`
Perform MOF Reconstruction for 2D with optional weights. Defaults to even weighting.
- template<class CellType >
`PlanarSeparator reconstructionWithMOF3D (const CellType &a_cell, const SeparatedMoments< VolumeMoments > &a_svm, const double a_internal_weight=0.5, const double a_external_weight=0.5)`
Perform MOF Reconstruction for 3D with optional weights. Defaults to even weighting.
- template<class MomentsContainerType , class CellType >
`PlanarSeparator reconstructionWithAdvectionNormals (const MomentsContainerType &a_volume_moments_list, const R2PNeighborhood< CellType > &a_neighborhood, const double a_two_plane_threshold=0.95)`
Reconstruct a `PlanarSeparator` from a collection of `VolumeMomentsAndNormal` objects, with supplying the threshold for when one or two planes are used.
- template<class CellType >
`PlanarSeparator reconstructionWithR2P2DDebug (const R2PNeighborhood< CellType > &a_neighborhood_geometry, PlanarSeparator a_initial_reconstruction)`
Perform R2P reconstruction for a 2D problem in the x-y plane and prints out all accepted reconstructions on path to best solution.
- template<class CellType >
`PlanarSeparator reconstructionWithR2P3DDebug (const R2PNeighborhood< CellType > &a_neighborhood_geometry, PlanarSeparator a_initial_reconstruction)`
Perform R2P reconstruction for a 3D problem and prints out all accepted reconstructions on path to best solution.
- `PlanarSeparator reconstructionWithELVIRA2DDebug (const ELVIRANeighborhood &a_neighborhood_geometry)`
Perform ELVIRA Reconstruction for 2D and prints out each of attempted reconstructions.
- `PlanarSeparator reconstructionWithELVIRA3DDebug (const ELVIRANeighborhood &a_neighborhood_geometry)`
Perform ELVIRA Reconstruction for 3D and prints out each of attempted reconstructions.
- template<class CellType >
`PlanarSeparator reconstructionWithLVIRA2DDebug (const LVIRANeighborhood< CellType > &a_neighborhood_geometry, PlanarSeparator a_initial_reconstruction)`
Perform LVIRA reconstruction for a 2D problem in the x-y plane and prints out all accepted reconstructions on path to best solution.
- template<class CellType >
`PlanarSeparator reconstructionWithLVIRA3DDebug (const LVIRANeighborhood< CellType > &a_neighborhood_geometry, PlanarSeparator a_initial_reconstruction)`
Perform LVIRA reconstruction for a 3D problem and prints out all accepted reconstructions on path to best solution.
- template<class CellType >
`PlanarSeparator reconstructionWithMOF2DDebug (const CellType &a_cell, const SeparatedMoments< VolumeMoments > &a_svm, const double a_internal_weight=0.5, const double a_external_weight=0.5)`
Perform MOF Reconstruction for a 2D problem and prints out all accepted reconstructions on path to best solution.
- template<class CellType >
`PlanarSeparator reconstructionWithMOF3DDebug (const CellType &a_cell, const SeparatedMoments< VolumeMoments > &a_svm, const double a_internal_weight=0.5, const double a_external_weight=0.5)`
Perform MOF Reconstruction for a 3D problem and prints out all accepted reconstructions on path to best solution.
- template<class MomentsContainerType , class CellType >
`PlanarSeparator reconstructionWithAdvectionNormalsDebug (const MomentsContainerType &a_volume_moments_list, const R2PNeighborhood< CellType > &a_neighborhood, const double a_two_plane_threshold=0.95)`
Reconstruct a `PlanarSeparator` from a collection of `VolumeMomentsAndNormal` objects, with supplying the threshold for when one or two planes are used.

- template<class CellType , class PlanarType >
`void setDistanceToMatchVolumeFraction` (const CellType &a_cell, const double a_volume_fraction, PlanarType *a_reconstruction, const double a_volume_fraction_tolerance=global_constants::TWO_PLANE_DISTANCE_VOLUME_FRACTION_TOLERANCE)
 - template<class CellType , class PlanarType >
`void setDistanceToMatchVolumeFractionPartialFill` (const CellType &a_cell, const double a_volume_fraction, PlanarType *a_reconstruction, const double a_volume_fraction_tolerance=global_constants::TWO_PLANAR_DISTANCE_VOLUME_FRACTION_TOLERANCE)
- Sets distance in a_reconstruction to result in given liquid volume fraction.*
- template<class PlanarType >
`void setDistanceToMatchVolumeFractionPartialFill` (const RectangularCuboid &a_cell, const double a_volume_fraction, PlanarType *a_reconstruction, const double a_volume_fraction_tolerance=global_constants::TWO_PLANE_DISTANCE_VOLUME_FRACTION_TOLERANCE)
- Specialization for RectangularCuboids that calls Analytical distance finding if a single plane.*
- template<class VolumeMomentsType >
`AccumulatedVolumeMoments< VolumeMomentsType > operator*` (const AccumulatedVolumeMoments< VolumeMomentsType > &a_list, const double a_multiplier)
- Overload * operator to multiply the contained VolumeMomentsType.*
- template<class VolumeMomentsType >
`AccumulatedVolumeMoments< VolumeMomentsType > operator*` (const double a_multiplier, const AccumulatedVolumeMoments< VolumeMomentsType > &a_list)
- Overload * operator to multiply the contained VolumeMomentsType.*
- template<class VolumeMomentsType >
`std::ostream & operator<<` (std::ostream &out, const ListedVolumeMoments< VolumeMomentsType > &a_listed_volume_moments)
 - template<class MomentsType >
`std::ostream & operator<<` (std::ostream &out, const SeparatedMoments< MomentsType > &a_separated_volume_moments)
 - template<class MomentsType >
`SeparatedMoments< MomentsType > operator*` (const SeparatedMoments< MomentsType > &a_svm, const double a_multiplier)
- Overload * operator to multiply the two geometric moments.*
- template<class MomentsType >
`SeparatedMoments< MomentsType > operator*` (const double a_multiplier, const SeparatedMoments< MomentsType > &a_svm)
- Overload * operator to multiply the two geometric moments.*
- template<class VolumeMomentsType >
`std::ostream & operator<<` (std::ostream &out, const TaggedAccumulatedVolumeMoments< VolumeMomentsType > &a_list)
 - template<class VolumeMomentsType >
`TaggedAccumulatedVolumeMoments< VolumeMomentsType > operator*` (const TaggedAccumulatedVolumeMoments< VolumeMomentsType > &a_list, const double a_multiplier)
- Overload * operator to multiply the contained VolumeMomentsType.*
- template<class VolumeMomentsType >
`TaggedAccumulatedVolumeMoments< VolumeMomentsType > operator*` (const double a_multiplier, const TaggedAccumulatedVolumeMoments< VolumeMomentsType > &a_list)
- Overload * operator to multiply the contained VolumeMomentsType.*
- std::ostream & `operator<<` (std::ostream &out, const Volume &a_volume)
 - std::ostream & `operator<<` (std::ostream &out, const VolumeMoments &a_volume_moments)
 - `VolumeMoments operator+` (const VolumeMoments &a_vm1, const VolumeMoments &a_vm2)
- Overload + operator to add two geometric moments together.*
- `VolumeMoments operator-` (const VolumeMoments &a_vm1, const VolumeMoments &a_vm2)
- Overload - operator to subtract one geometric moment object from another.*
- `VolumeMoments operator*` (const double a_multiplier, const VolumeMoments &a_vm)
- Overload * operator to multiply volume/centroid.*

- `VolumeMoments operator* (const VolumeMoments &a_vm, const double a_multiplier)`
*Overload * operator to multiply volume/centroid.*
- template<UnsignedIndex_t kArrayLength>
`std::ostream & operator<< (std::ostream &out, const VolumeMomentsAndDoubles<kArrayLength> &a<->_volume_moments_and_doubles)`
- template<UnsignedIndex_t kArrayLength>
`VolumeMomentsAndDoubles<kArrayLength> operator+ (const VolumeMomentsAndDoubles<kArrayLength> &a_vmad1, const VolumeMomentsAndDoubles<kArrayLength> &a_vmad2)`
- template<UnsignedIndex_t kArrayLength>
`VolumeMomentsAndDoubles<kArrayLength> operator- (const VolumeMomentsAndDoubles<kArrayLength> &a_vmad1, const VolumeMomentsAndDoubles<kArrayLength> &a_vmad2)`
- template<UnsignedIndex_t kArrayLength>
`VolumeMomentsAndDoubles<kArrayLength> operator*> (const double a_multiplier, const VolumeMomentsAndDoubles<kArrayLength> &a_vmad)`
- template<UnsignedIndex_t kArrayLength>
`VolumeMomentsAndDoubles<kArrayLength> operator*> (const VolumeMomentsAndDoubles<kArrayLength> &a_vmad, const double a_multiplier)`
- std::ostream & operator<< (std::ostream &out, const VolumeMomentsAndNormal &a_volume_moments_and_normal)
- void `setMinimumVolumeToTrack (const double a_minimum_volume_to_track)`
Function to set MINIMUM_VOLUME_TO_TRACK
- void `setMinimumSurfaceAreaToTrack (const double a_minimum_surface_area_to_track)`
Function to set MINIMUM_SURFACE_AREA_TO_TRACK
- void `setVolumeFractionBounds (const double a_VF_low)`
Set VF_LOW and VF_HIGH defined in src/constants.cpp.
- void `setVolumeFractionTolerance (const double a_tolerance)`
Set the volume fraction tolerance for iterative distance finding.
- std::ostream & operator<< (std::ostream &out, const NullReconstruction &a_reconstruction)
- std::ostream & operator<< (std::ostream &out, const PlanarLocalizer &a_reconstruction)
- template<UnsignedIndex_t kStackPlanes>
`std::ostream & operator<< (std::ostream &out, const PlanarReconstruction<kStackPlanes> &a<->reconstruction)`
- std::ostream & operator<< (std::ostream &out, const PlanarSeparator &a_reconstruction)
- template<class ReconstructionType, template<class NodeType> class GraphType>
`std::ostream & operator<< (std::ostream &out, const ReconstructionLink<ReconstructionType, GraphType> &a_reconstruction)`

Variables

- static IRL::c_RuntimeCuttingMethod **C_CUTTING_METHOD**
- const **NullReconstruction** & **a_reconstruction**
- HalfEdgePolytopeType * **a_complete_polytope**
- const **MathVector<kNumberOfElements>** & **a_vec_1**
- const **MathVector<kNumberOfElements>** & **a_vec**
- const double **a_double**
- const **RectangularCuboid** **unit_cell**
- static constexpr UnsignedIndex_t **LVIRA_2D_columns** = 1
Number of columns for 2D LVIRA optimization, which is equal to parameters being fit.
- static constexpr UnsignedIndex_t **LVIRA_3D_columns** = 2
Number of columns for 3D LVIRA optimization, which is equal to parameters being fit.
- static constexpr UnsignedIndex_t **R2P_2D1P_ncells** = 9
Number of cells for 2D - 1 plane R2P optimization.
- static constexpr UnsignedIndex_t **R2P_2D1P_rows** = 7 * **R2P_2D1P_ncells** + 1

- static constexpr UnsignedIndex_t [R2P_2D1P_columns](#) = 1

Number of columns for 2D - 1 plane R2P optimization, which is equal to parameters being fit.
- static constexpr UnsignedIndex_t [R2P_3D1P_ncells](#) = 27

Number of cells for 3D - 1 plane R2P optimization.
- static constexpr UnsignedIndex_t [R2P_3D1P_rows](#) = 7 * [R2P_3D1P_ncells](#) + 1

Number of rows for 3D - 1 plane R2P optimization, with 7 Entries per cell (Liquid volume, liquid centroid, gas centroid) and the surface area for the cell being reconstructed.
- static constexpr UnsignedIndex_t [R2P_3D1P_columns](#) = 2

Number of columns for 3D - 1 plane R2P optimization, which is equal to parameters being fit.
- static constexpr UnsignedIndex_t [R2P_2D2P_ncells](#) = 9

Number of cells for 2D - 2 plane R2P optimization.
- static constexpr UnsignedIndex_t [R2P_2D2P_rows](#) = 7 * [R2P_2D2P_ncells](#) + 1

Number of rows for 2D - 2 plane R2P optimization, with 7 Entries per cell (Liquid volume, liquid centroid, gas centroid) and the surface area for the cell being reconstructed.
- static constexpr UnsignedIndex_t [R2P_2D2P_columns](#) = 4

Number of columns for 2D - 2 plane R2P optimization, which is equal to parameters being fit.
- static constexpr UnsignedIndex_t [R2P_3D2P_ncells](#) = 27

Number of cells for 3D - 2 plane R2P optimization.
- static constexpr UnsignedIndex_t [R2P_3D2P_rows](#) = 7 * [R2P_3D2P_ncells](#) + 1

Number of rows for 3D - 2 plane R2P optimization, with 7 Entries per cell (Liquid volume, liquid centroid, gas centroid) and the surface area for the cell being reconstructed.
- static constexpr UnsignedIndex_t [R2P_3D2P_columns](#) = 6

Number of columns for 3D - 2 plane R2P optimization, which is equal to parameters being fit.

5.38.1 Detailed Description

Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Decomposition of geometric objects into simplices that are then intersected with planes in reconstructions to generate groups of simplices that lay solely on one side of the plane.

This cutting takes an original object, such as Tetrahedron or Dodecahedron and applies its `setHalfEdgeVersion` method to on a static allocation of a [HalfEdgePolygon](#) or [HalfEdgePolyhedron](#). This is done through lazy evaluation in the function `setHalfEdgeStructure`, since the memory addresses for each HalfEdge/VertexLocation/ \leftarrow Vertex/Face will be the same, with only the `VertexLocation`'s `Pt` changing. A [SegmentedHalfEdgePolygon](#) or [SegmentedHalfEdgePolyhedron](#) is then generated from this, which is just a collection of vertices and faces from this over-arching "complete" `HalfEdgePolygon`/`HalfEdgePolyhedron`. This is then cut by the planar reconstruction objects through the call to `getVolumeMoments(...)`, where a pointer to the Segmented and complete polytope are provided, along with the reconstruction. If only the area underneath the reconstruction is needed, the `SegmentedPolytope` is changed in place, where the Faces and vertices are changed, with new ones added, representing the new half-edge structure of the object that lays completely underneath the reconstructions. If instead the objects on both sides of the plane are needed, such as during linked cutting where volumes are distributed across a system of reconstructions with known representations, then new [SegmentedHalfEdgePolytope](#) objects are generated to represent the clipped (above plane) portions. During the cutting of these half edge structures, all new `HalfEdge`/ \leftarrow `VertexLocation`/`Vertex`/`Face` objects are taken from the allocation of the original [HalfEdgePolytope](#), and therefore this must have stable memory so that pointers remain valid during cutting. It is for this reason that the original [HalfEdgePolytope](#) is referred to as "complete", since at the end it will contain the complete polytope, subdivided amongst the reconstructions, an accumulation of the Segmented versions generated during cutting. Note, since Segmented versions essentially store all data on the original [HalfEdgePolytope](#), with just pointers to the data it owns there, the Segmented versions can be made invalid if the `complete_polytope` is modified directly, so it's best not to do this if `SegmentedPolytopes` you hope to use still exist.

This form of cutting relies on the ability to separate an object into a collection of signed simplices. In the context of IRL, each object should have the methods `getNumberOfSimplicesInDecomposition()`, which is the number of signed simplices that are needed to represent the original object, and `getSimplexFromDecomposition(UnsignedIndex_t)`, which returns one of the simplices. This simplex is then recursively divided by the reconstruction into more simplices that lay above and below the planes in the reconstruction, ultimately being left with simplices that are either below or above the reconstruction. Once that is the case, the moments for the simplices are then calculate and accumulated in the `volume_moments` variable. What happens to the newly generated simplices is dictated by the reconstruction type and type of moments that need to be returned. These different behaviors are created through the use of common static interfaces for treating the simplices, mainly defined in `cut_simplex.h`, which specializes the class in `cut_simplex_drivers.h`. Ultimately, there are two processes that happen. The simplex is separated into two by a plane and new simplices above and below it can be generated using the lookup tables in `lookup.h`. The simplices underneath the plane in the reconstruction continue to be divided until no planes are left, at which time it is treated according to a function in `handle_enclosed_volume.h`. Volumes above the planes are treated through functins in `continue_dividing_volume.h`, which handles things such as distributing these volumes over cutting networks or simply ignoring the volumes above the planes. The special cases of 2D simplices (triangles) or 3D simplices (tetrahedra) are handled through the `SimplexWrapper<SimplexType>`, which is a specialized struct given unique behaviors to certain functions depending on the dimension. These simplex wrappers are conveniently described in `simplex_wrappers.h`

5.38.2 Function Documentation

5.38.2.1 __attribute__()

```
template<UnsignedIndex_t kNumberOfElements>
IRL::__attribute__ (
    (const) ) const &
```

Overload * operator to be dot product of the two vectors.

Overload / operator between double and `MathVector` to be return a `MathVector`.

Overload * operator between double and `MathVector` to be return a `MathVector`.

Overload - operator between `MathVectors` of same length to add elements.

Overload + operator between `MathVectors` of same length to add elements.

5.38.2.2 checkPureObject()

```
template<class EncompassingType , class ReconstructionType >
int IRL::checkPureObject (
    const EncompassingType & a_encompassing_polyhedron,
    const ReconstructionType & a_reconstruction ) [inline]
```

Test if the object passed to it lays entirely on one side of the reconstruction.

This function returns whether the polyhedron lays entirely on one side of the reconstruction. Return values are as followed:

- 0 : Returned if polyhedron lays entirely internal to the reconstruction (under the plane).
- 1 : Returned if polyhedron lays entirely external to the reconstruction (over the plane).
- 2 : Returned if the reconstruction intersects the polyhedron.

5.38.2.3 crossProduct()

```
template<class DataType >
DataType IRL::crossProduct (
    const DataType & a_vector_0,
    const DataType & a_vector_1 ) [inline]
```

Take cross product of two 3-element vectors.

Template requirements for DataType:

- Overload `operator[]` : Method that supplies access to the underlying variable to use for calculating magnitude.

5.38.2.4 crossProductNormalized()

```
template<class DataType >
DataType IRL::crossProductNormalized (
    const DataType & a_vector_0,
    const DataType & a_vector_1 ) [inline]
```

Cross product of two 3-element vectors then normalized.

Template requirements for DataType:

- Overload `operator[]` : Method that supplies access to the underlying variable that cross products will be taken against.
- Overload `operator/=` : Method that divides each of its three components by a double to normalize the vector.

5.38.2.5 cutPlaneByHexahedron()

```
template<class PolygonType , class HexahedronType >
PolygonType IRL::cutPlaneByHexahedron (
    const HexahedronType & a_hexahedron,
    const Plane & a_plane )
```

Cuts a plane by `a_hexahedron`, returning a `PlanePolygon` which is a polygon of the plane contained inside the `rectangular_cuboid`.

NOTE: This function does not set the centroid stored in a `DividedPolygon` when `PolygonType` is `DividedPolygon`.

Template requirements for `HexahedronType`:

- Must be a `Hexahedron` or `RectangularCuboid` `class` object.

Parameters

in	<i>a_hexahedron</i>	Rectangular cuboid that <i>a_plane</i> will be applied to.
in	<i>a_plane</i>	Plane that will be cut by <i>a_hexahedron</i> .

5.38.2.6 *cutPolygonByPlane()*

```
template<class PolygonType >
PolygonType IRL::cutPolygonByPlane (
    const PolygonType & a_polygon,
    const Plane & a_cutting_plane,
    const double a_flip_cut )
```

Cuts a [PlanePolygon](#) by a given plane, returning a new Polygon of type [PolygonType](#) which is a polygon of the plane after being intersected with *a_cutting_plane*.

NOTE: This function does not set the centroid stored in a [DividedPolygon](#) when [PolygonType](#) is [DividedPolygon](#).

Template requirements for [PolygonType](#):

- Must be either [Polygon](#) or [DividedPolygon](#).

Parameters

in	<i>a_polygon</i>	Plane that will be cut by <i>a_cutting_plane</i> .
in	<i>a_cutting_plane</i>	Plane that will cut <i>a_plane_polygon</i> .
in	<i>a_flip_cut</i>	Indicates whether below planes (if 1.0) or above planes (if -1.0) polygon should be returned.

5.38.2.7 *dotProduct()*

```
template<class DataType >
double IRL::dotProduct (
    const DataType & a_vector_0,
    const DataType & a_vector_1 ) [inline]
```

Dot product between two 3 element vectors.

Template requirements for [DataType](#):

- Overload `operator[]` : Method that supplies access to the underlying variable that cross products will be taken against.

5.38.2.8 findDistanceOnePlane()

```
double IRL::findDistanceOnePlane (
    const RectangularCuboid & a_rectangular_cuboid,
    const double a_volume_fraction,
    const Normal & a_normal )
```

Volume conserving distance-finding routine for single-plane reconstructions.

This function uses the analytical calculation from Scardovelli & Zaleski, JCP 164,228-247 (2000) to calculate the distance to a single plane to result in a reconstruction with a given volume fraction. In this function, the main worker is `getAlpha(...)`.

Parameters

in	<code>a_rectangular_cuboid</code>	Cell for which the reconstruction with a plane of <code>a_normal</code> will be for.
in	<code>a_volume_fraction</code>	Liquid volume fraction the reconstruction should result in.
in	<code>a_normal</code>	<code>Normal</code> for a plane that needs the reconstruction.

5.38.2.9 getGeometricCaseId()

```
template<class DistanceListType >
LookupIndex_t IRL::getGeometricCaseId (
    const DistanceListType & a_distances ) [inline]
```

Given an array of distances to an intersecting plane and the length of that array, a unique integer ID is calculated based on the sign and index of each element in `a_distances`. This is often used to then reference unique cases in a lookup table. The ID is calculated as $\sum_{i=0}^{i=a_number_of_distances} 2^i * (0.5 + \text{sign}(0.5, a_distances[i]))$.

Parameters

in	<code>a_distances</code>	Array of signed distances for the vertices.
in	<code>a_number_of_distances</code>	Length of the array <code>a_distances</code> .

5.38.2.10 getPlanePolygonFromReconstruction()

```
template<class PolygonType , class PolyhedronType >
PolygonType IRL::getPlanePolygonFromReconstruction (
    const PolyhedronType & a_polyhedron,
    const PlanarSeparator & a_reconstruction,
    const Plane & a_plane_to_make_polygon )
```

Return ConvexPolygon that represents the plane in `a_reconstruction[a_plane_to_make_polygon]` on ``a_hexahedron`.

NOTE: This function does not set the centroid stored in a DividedPolygon when PolygonType is DividedPolygon.

Template requirements for HexahedronType:

- Must be a Hexahedron or RectangularCuboid class object.

Template requirements for PolygonType:

- Must be either Polygon or DividedPolygon.

Parameters

in	<i>a_hexahedron</i>	Rectangular cuboid a_reconstruction will be applied on.
in	<i>a_reconstruction</i>	Reconstruction to get plane from
in	<i>a_plane_to_make_polygon</i>	Plane index in a_reconstruction that will be made into a polygon.

5.38.2.11 getReconstructionSurfaceArea()

```
template<class HexahedronType >
double IRL::getReconstructionSurfaceArea (
    const HexahedronType & a_hexahedron,
    const PlanarSeparator & a_reconstruction )
```

Cut a supplied rectangular cuboid by a given reconstruction and return the surface area.

This function cuts a_hexahedron by a_reconstruction to obtain the interfacial surface area in that cell. This makes heavy use of large lookup tables held in `lookup_tables.h`.

NOTE: If a_reconstruction contains multiple planes, it will also cut the planes by one another before calculating surface area.

Template requirements for HexahedronType:

- Must be a Hexahedron or RectangularCuboid class object.

Template requirements for PolygonType:

- Must be either Polygon or DividedPolygon.

Parameters

in	<i>a_hexahedron</i>	Rectangular cuboid (such as a computational cell) that the reconstruction will be used for.
in	<i>a_reconstruction</i>	Planar reconstruction that will be cut by a_hexahedron and have its surface area calculated.

5.38.2.12 isPlaneIntersectingCell()

```
template<class GeometryType >
bool IRL::isPlaneIntersectingCell (
    const Plane & a_plane,
    const GeometryType & a_geometry ) [inline]
```

Checks to see if a plane intersects a given cell.

Template requirements for GeometryType:

- Overload `operator[]` : Method that supplies access to the underlying vertices as `Pts`.
- `int getNumberOfVertices(void)`: Method that returns the number of points in the geometry
`static const int max_number_of_vertices`: Maximum number of pts that can be in the geometry, used for allocation of a double array.

Parameters

in	<code>a_plane</code>	Plane that may or may not intersect <code>a_geometry</code>
in	<code>a_geometry</code>	Geometry on which to check if plane exists.

5.38.2.13 isPtInternal()

```
template<class ReconstructionType >
bool IRL::isPtInternal (
    const Pt & a_pt,
    const ReconstructionType & a_reconstruction ) [inline]
```

Checks to see if point is internal to `PlanarReconstruction`.

This function determines if the point `a_pt` is internal to the planar reconstruction or not. In the case of a single plane in `a_reconstruction`, this would mean if the point is underneath the plane.

Template requirements for ReconstructionType:

- Must be inherited from `PlanarReconstruction`, such as a `PlanarLocalizer` or `PlanarSeparator`.
- `double flip(void)` : Returns -1.0 if cutting is flipped (concave object) or 1.0 if cutting is not flipped (convex).

Parameters

in	<code>a_pt</code>	Point for which it will be determined if it is internal or not to <code>a_reconstruction</code> .
in	<code>a_reconstruction</code>	Reconstruction forming the volume <code>a_pt</code> may or may not be internal to.

5.38.2.14 magnitude()

```
template<class DataType >
double IRL::magnitude (
    const DataType & a_vector_0,
    const DataType & a_vector_1 ) [inline]
```

Calculate the magnitude between two 3 element vectors.

Template requirements for `DataType`:

- Overload `operator[]` : Method that supplies access to the underlying variable to use for calculating magnitude.

5.38.2.15 rotateNormalOntoNormal()

```
UnitQuaternion IRL::rotateNormalOntoNormal (
    const Normal & a_normal_original,
    const Normal & a_normal_destination,
    double * a_rotation_amount,
    Normal * a_rotation_axis )
```

Return the quaternion that would rotate `a_normal_original` onto `a_normal_destination`.

This function returns a quaternion that will rotate the normal `a_normal_original` onto the normal `a_normal_destination`. Additionally, the rotation amount (in radians) and the rotation axis will be given through the pointers `a_rotation_amount` and `a_rotation_axis`, respectively.

Parameters

in	<code>a_normal_original</code>	Original normal we want to rotate.
in	<code>a_normal_destination</code>	Normal we want original normal to become from rotation.
out	<code>a_rotation_amount</code>	Amount of rotation used to make the returned <code>UnitQuaternion</code> .
out	<code>a_rotation_axis</code>	Rotation axis used to make the returned <code>UnitQuaternion</code> .

5.38.2.16 scalarTripleProduct()

```
template<class DataType >
double IRL::scalarTripleProduct (
    const DataType & a_vector_0,
    const DataType & a_vector_1,
    const DataType & a_vector_2 ) [inline]
```

Scalar triple product of 3, 3 element vectors.

Template requirements for `DataType`:

- Overload `operator[]` : Method that supplies access to the underlying variable that cross products will be taken against.

5.38.2.17 `setDistanceToMatchVolumeFractionPartialFill()`

```
template<class CellType , class PlanarType >
void IRL::setDistanceToMatchVolumeFractionPartialFill (
    const CellType & a_cell,
    const double a_volume_fraction,
    PlanarType * a_reconstruction,
    const double a_volume_fraction_tolerance = global_constants::TWO_PLANE_DISTANCE_↔
VOLUME_FRACTION_TOLERANCE ) [inline]
```

Sets distance in `a_reconstruction` to result in given liquid volume fraction.

This function sets the distance in `a_reconstruction` to lead to the liquid volume fraction `a_volume_↔fraction` in the unit computational cell $[-0.5, 0.5]^3$. The distance that will be set will be in reference to the center of the unit cell as 0.0. If different coordinate system is needed, make sure to correct after use. If a one-plane reconstruction, the volume fraction will be recreated exactly through an analytical calculation. If a two-plane reconstruction the volume fraction will be recreated to within `a_volume_fraction_tolerance` using Newton-Raphson optimization and potentially bisection.

Parameters

in	<code>a_reconstruction</code>	Pointer to PlanarSeparator object that will have its plane's distances set.
in	<code>a_rectangular_cuboid</code>	Cell for which the reconstruction with a plane of <code>a_normal</code> will be for. Currently only used for single-plane reconstruction. Two plane reconstruction still assumed to be on unit cube.
in	<code>a_volume_fraction</code>	Liquid volume fraction that the updated distance will recreate.
in	<code>a_volume_fraction_tolerance</code>	Tolerance allowed in recreating <code>a_volume_fraction</code> .

5.38.2.18 `setVolumeFractionBounds()`

```
void IRL::setVolumeFractionBounds (
    const double a_VF_low )
```

Set VF_LOW and VF_HIGH defined in `src/constants.cpp`.

This function sets VF_LOW for all future computations. VF_LOW is used as a lower threshold to terminate some computations, especially during calculations of interface reconstructions. VF_HIGH will automatically be set as $1.0 - a_VF_low$ in order to preserve symmetry.

Parameters

in	<code>a_VF_low</code>	Value to set VF_LOW.
----	-----------------------	----------------------

5.38.2.19 setVolumeFractionTolerance()

```
void IRL::setVolumeFractionTolerance (
    const double a_tolerance )
```

Set the volume fraction tolerance for iterative distance finding.

This function sets the volume fraction tolerance to be used when an iterative distance finding routine is used. It will always be the minimum of `a_tolerance` and `VF_LOW`.

Parameters

<code>in</code>	<code>a_tolerance</code>	Volume fraction tolerance to use during iterative distance finding.
-----------------	--------------------------	---

5.38.2.20 signedDistanceToPolygonVertices()

```
template<class PolygonType >
void IRL::signedDistanceToPolygonVertices (
    const PolygonType & a_polygon,
    const Plane & a_cutting_plane,
    const double a_flip_cut,
    std::array< double, 6 > * a_distance_to_vertices ) [inline]
```

Loops through the vertices of a polygon and computes a signed distance from `a_cutting_plane` for each. Also sets all remaining distances in array to distance value of first vertex to be cyclic.

This function computes the signed distance from `a_cutting_plane` to the vertices in `a_polygon` using the `signedDistanceToPoint` method of `a_cutting_plane`. `flip_cut` is needed in order to flip the sign of the distance and cut for the correct phase when the reconstruction is comprised of multiple planes. A supplied array of `double[6]` is modified in the function.

Parameters

<code>in</code>	<code>a_polygon</code>	Polygon needing signed distances for its vertices.
<code>in</code>	<code>a_cutting_plane</code>	The plane that is being intersected with the polygon.
<code>in</code>	<code>a_flip_cut</code>	Whether the sign should be flipped (indicating cutting to obtain quantities for above the plane instead of below).
<code>out</code>	<code>a_distance_to_vertices</code>	Array of <code>double[6]</code> to place computed distances for the vertices. to the vertices.

5.38.2.21 signedDistanceToVertices()

```
template<class GeometryType , class VertexDistanceList >
void IRL::signedDistanceToVertices (
    const GeometryType & a_geometry,
    const Plane & a_cutting_planem,
    VertexDistanceList * a_distance_to_vertices ) [inline]
```

Loops through the vertices of the supplied geometry and computes a signed distance from `a_cutting_plane` for each.

This function computes the signed distance from `a_cutting_plane` to the vertices in `a_geometry` using the `signedDistanceToPoint` method of `a_cutting_plane`. `flip_cut` is needed in order to flip the sign of the distance and cut for the correct phase when the reconstruction is comprised of multiple planes. A supplied array of `double[]` is modified in the function.

Template requirements for `GeometryType`:

- Overload `operator[]` : Method that supplies access to the underlying vertices as `Pts`s.
- `int getNumberOfVertices(void)` : Method that returns the number of points in the geometry

Parameters

<code>in</code>	<code>a_tet</code>	The tet needing signed distances for its vertices.
<code>in</code>	<code>a_cutting_plane</code>	The plane that is being intersected with the tet.
<code>in</code>	<code>flip_cut</code>	Whether the sign should be flipped (indicating cutting to obtain quantities for above the plane instead of below).
<code>out</code>	<code>a_distance_to_vertices</code>	Array of <code>double[GeometryType::max_number_of_vertices]</code> to place computed distance for vertices.

5.38.2.22 `sortAscendingBasedOnOtherArray()`

```
template<class CarriedType , class DictatingType >
void IRL::sortAscendingBasedOnOtherArray (
    CarriedType * a_carried_array,
    DictatingType * a_dictating_array ) [inline]
```

This function sorts `a_carried_array` based on the elements in `a_dictating_array` using insertion sort.

Template requirements for `CarriedType`:

- Access via overloaded `operator[]`.
- Ability to be used with `std::swap()`.

Template requirements for `DictatingType`:

- Access via overloaded `operator[]`.
- Ability to be used with `std::swap()`.
- Comparison operator `>(int i1, int i2)` that returns whether one the element at `i1` is greater than the element at `i2`.

Parameters

in, out	<i>a_carried_array</i>	Array that will be sorted into ascending order according to the elements in <i>a_dictating_array</i>
in, out	<i>a_dictating_array</i>	Array that will be sorted and drive the sorting in <i>a_carried_array</i>
in	<i>a_length_of_array</i>	Length of both arrays being sorted.

5.38.2.23 sortDescendingBasedOnOtherArray()

```
template<class CarriedType , class DictatingType >
void IRL::sortDescendingBasedOnOtherArray (
    CarriedType * a_carried_array,
    DictatingType * a_dictating_array ) [inline]
```

This function sorts *a_carried_array* based on the elements in *a_dictating_array* using insertion sort.

Template requirements for CarriedType:

- Access via overloaded operator[].
- Ability to be used with std::swap().

Template requirements for DictatingType:

- Access via overloaded operator[].
- Ability to be used with std::swap().
- Comparison operator <(int i1, int i2) that returns whether one the element at i1 is less than the element at i2.

Parameters

in, out	<i>a_carried_array</i>	Array that will be sorted into descending order according to the elements in <i>a_dictating_array</i>
in, out	<i>a_dictating_array</i>	Array that will be sorted and drive the sorting in <i>a_carried_array</i>
in	<i>a_length_of_array</i>	Length of both arrays being sorted.

5.38.2.24 squaredMagnitude()

```
template<class DataType >
double IRL::squaredMagnitude (
    const DataType & a_vector ) [inline]
```

Calculate the squared magnitude between for a 3 element vector.

Template requirements for DataType:

- Overload operator[] : Method that supplies access to the underlying variable that cross products will be taken against.

5.38.3 Variable Documentation

5.38.3.1 C_CUTTING_METHOD

```
IRL::c_RuntimeCuttingMethod IRL::C_CUTTING_METHOD [static]
```

Initial value:

```
= c_getCompiledDefaultCuttingMethod<DefaultCuttingMethod>()
```

5.38.3.2 unit_cell

```
const RectangularCuboid IRL::unit_cell
```

Initial value:

```
= RectangularCuboid::fromBoundingPts(Pt(-0.5, -0.5, -0.5), Pt(0.5, 0.5, 0.5))
```

5.39 irl_fortran_interface Module Reference

This is just a master wrapper for the entire [IRL](#) fortran interface. For information about each module, view the documentation for the module itself.

5.39.1 Detailed Description

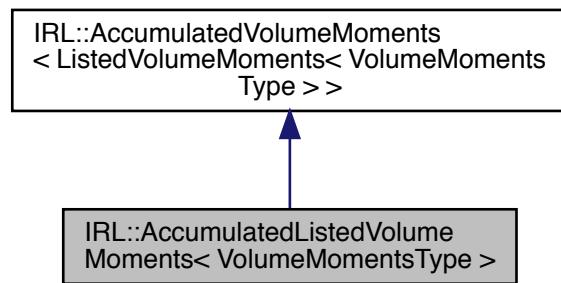
This is just a master wrapper for the entire [IRL](#) fortran interface. For information about each module, view the documentation for the module itself.

Chapter 6

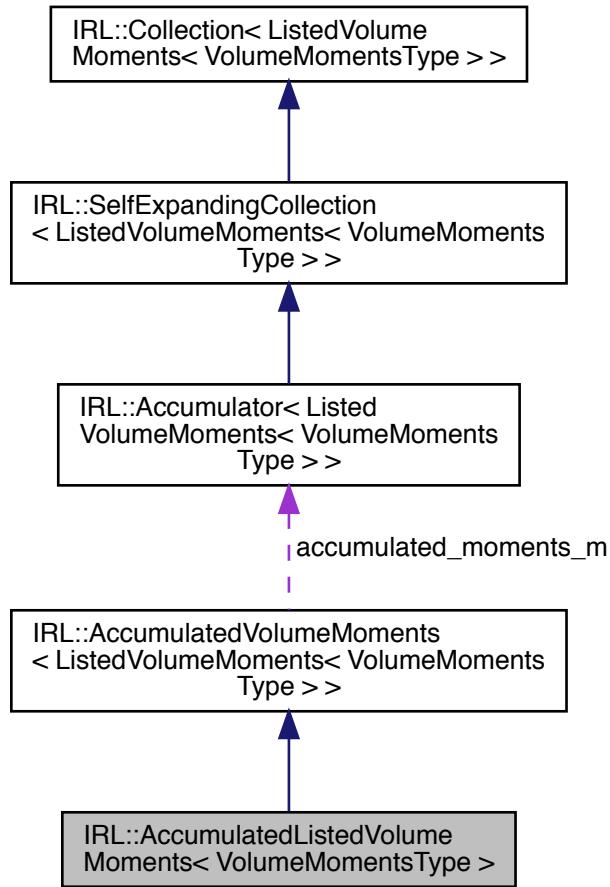
Class Documentation

6.1 IRL::AccumulatedListedVolumeMoments< VolumeMomentsType > Class Template Reference

Inheritance diagram for IRL::AccumulatedListedVolumeMoments< VolumeMomentsType >:



Collaboration diagram for IRL::AccumulatedListedVolumeMoments< VolumeMomentsType >:



Public Types

- using **contained_type** = VolumeMomentsType

Public Member Functions

- [AccumulatedListedVolumeMoments \(void\)=default](#)
Default constructor.
- [~AccumulatedListedVolumeMoments \(void\)=default](#)
Default destructor.

The documentation for this class was generated from the following file:

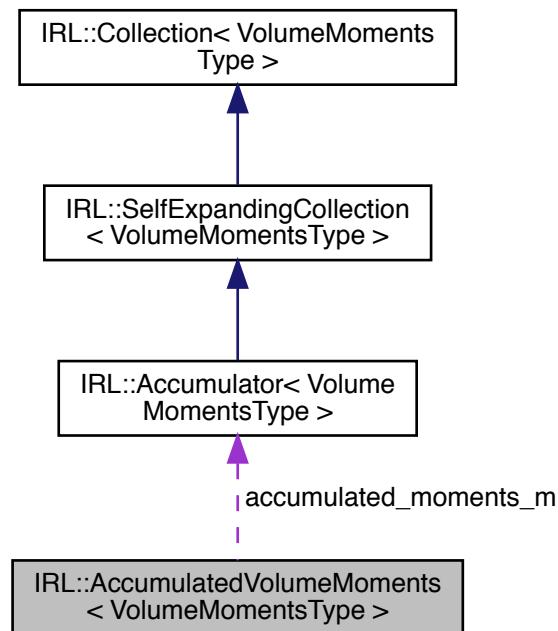
- accumulated_listed_volume_moments.h

6.2 IRL::AccumulatedVolumeMoments< VolumeMomentsType > Class Template Reference

Self-expanding vector of VolumeMoments.

```
#include <accumulated_volume_moments.h>
```

Collaboration diagram for IRL::AccumulatedVolumeMoments< VolumeMomentsType >:



Public Types

- using **iterator** = typename [Accumulator< VolumeMomentsType >::iterator](#)
- using **const_iterator** = typename [Accumulator< VolumeMomentsType >::const_iterator](#)
- using **contained_type** = [VolumeMomentsType](#)

Public Member Functions

- [AccumulatedVolumeMoments](#) (void)=default
Default constructor.
- UnsignedIndex_t [size](#) (void) const
Get size of the collection.
- contained_type & [operator\[\]](#) (const UnsignedIndex_t a_index)
This will self-expand to prevent itself from accessing out of bounds memory.
- const contained_type & [operator\[\]](#) (const UnsignedIndex_t a_index) const
Const version for access to object in collection.

- `AccumulatedVolumeMoments & operator= (const double a_value)`
Set all moments in the list equal to a_value.
- `void normalizeByVolume (void)`
Normalize entire vector by volume.
- `void multiplyByVolume (void)`
Normalize entire vector by volume.
- `AccumulatedVolumeMoments & operator+= (const AccumulatedVolumeMoments &a_rhs)`
Overload operator+= to accumulate moments with the same index, and extend to match a_rhs length.
- `void clear (void)`
Empty the container.
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept
- `~AccumulatedVolumeMoments (void)=default`
Default destructor.

Private Attributes

- `Accumulator< VolumeMomentsType > accumulated_moments_m`

6.2.1 Detailed Description

```
template<class VolumeMomentsType>
class IRL::AccumulatedVolumeMoments< VolumeMomentsType >
```

Self-expanding vector of `VolumeMoments`.

The documentation for this class was generated from the following file:

- `accumulated_volume_moments.h`

6.3 IRL::AccumulateIntoCollection Struct Reference

Static Public Member Functions

- `template<class ReturnType , class ReconstructionType , class SimplexType >`
`static void handleVolumeCompletelyInternalToPlanarReconstruction (const SimplexType &a_simplex,`
`const ReconstructionType &a_reconstruction, ReturnType *a_moments_to_return)`

The documentation for this struct was generated from the following file:

- `handle_enclosed_volume.h`

6.4 IRL::AccumulateIntoScalar Struct Reference

Static Public Member Functions

- template<class ReturnType , class ReconstructionType , class SimplexType >
static void **handleVolumeCompletelyInternalToPlanarReconstruction** (const SimplexType &a_simplex,
const ReconstructionType &a_reconstruction, ReturnType *a_moments_to_return)

The documentation for this struct was generated from the following file:

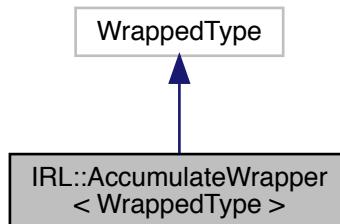
- handle_enclosed_volume.h

6.5 IRL::AccumulateWrapper< WrappedType > Class Template Reference

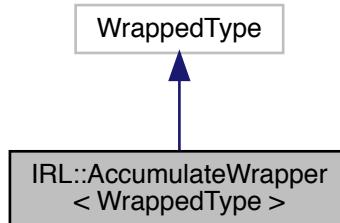
This is essentially a class that delays the evaluation of Tagged moments during generic_cutting with a [ReconstructionLink](#) for an eventual collection. As an example, this class needs to be used when cutting by a LocalizedSeparatorPath for tagged moments, since the first cutting operation (by the Localizer), is not what will dictate the tagging of the moments, but instead the second (by the PlanarSeparatorPath) does. The use of the tagged moments that are going to be returned need to be delayed and accumulated. We would then return them as auto moments = getVolumeMoments<[AccumulateWrapper](#)<TaggedAccumulatedVolumeMoments<Volume↔Moments>>> (a_volume_to_cut, a_LocalizedSeparatorPath);.

```
#include <accumulate_wrapper.h>
```

Inheritance diagram for IRL::AccumulateWrapper< WrappedType >:



Collaboration diagram for IRL::AccumulateWrapper< WrappedType >:



Public Types

- using **contained_type** = WrappedType

6.5.1 Detailed Description

```
template<class WrappedType>
class IRL::AccumulateWrapper< WrappedType >
```

This is essentially a class that delays the evaluation of Tagged moments during generic_cutting with a [ReconstructionLink](#) for an eventual collection. As an example, this class needs to be used when cutting by a LocalizedSeparatorPath for tagged moments, since the first cutting operation (by the Localizer), is not what will dictate the tagging of the moments, but instead the second (by the PlanarSeparatorPath) does. The use of the tagged moments that are going to be returned need to be delayed and accumulated. We would then return them as auto moments = getVolumeMoments<[AccumulateWrapper](#)<TaggedAccumulatedVolumeMoments<Volume->Moments>>> (a_volume_to_cut, a_LocalizedSeparatorPath);.

The documentation for this class was generated from the following file:

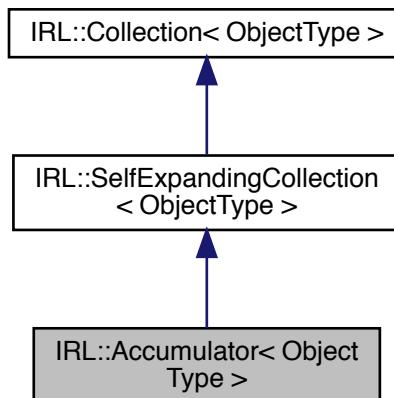
- accumulate_wrapper.h

6.6 IRL::Accumulator< ObjectType > Class Template Reference

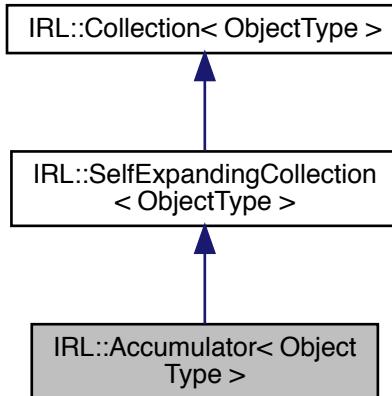
This class is a [SelfExpandingCollection](#) that has operator += call the += operator for each member in its collection.

```
#include <accumulator.h>
```

Inheritance diagram for IRL::Accumulator< ObjectType >:



Collaboration diagram for IRL::Accumulator< ObjectType >:



Public Member Functions

- **Accumulator** (void)=default
Default constructor.
- **Accumulator & operator+=** (const [Accumulator](#) &a_rhs)
The operator += will be used to add another [Accumulator](#), where the entry in each index is summed between the two.
- **~Accumulator** (void)=default
Default destructor.

Additional Inherited Members

6.6.1 Detailed Description

```
template<class ObjectType>
class IRL::Accumulator< ObjectType >
```

This class is a [SelfExpandingCollection](#) that has operator += call the += operator for each member in its collection.

The documentation for this class was generated from the following file:

- accumulator.h

6.7 f_lviraneighborhood_rectangularcuboid_class::addmember Interface Reference

Public Member Functions

- subroutine **lviraneighborhood_rectangularcuboid_class_addmember** (this, a_rectangular_cuboid, a_<volume_fraction>)

The documentation for this interface was generated from the following file:

- f_lviraneighborhood_rectangularcuboid_class.f90

6.8 f_r2pneighborhood_rectangularcuboid_class::addmember Interface Reference

Public Member Functions

- subroutine **r2pneighborhood_rectangularcuboid_class_addmember** (this, a_rectangular_cuboid, a_← separated_volume_moments)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.9 f_planarseparator_class::addplane Interface Reference

Public Member Functions

- subroutine **planarseparator_class_addplane** (this, a_normal, a_distance)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.10 f_planarlocalizer_class::addplane Interface Reference

Public Member Functions

- subroutine **planarlocalizer_class_addplane** (this, a_normal, a_distance)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.11 f_cappeddodecahedron_doubles3_class::adjustcaptomatchvolume Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_doubles3_class_adjustcaptomatchvolume** (this, a_correct_signed_← _volume)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.12 f_cappeddodecahedron_class::adjustcaptomatchvolume Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_class_adjustcaptomatchvolume** (this, a_correct_signed_volume)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.13 f_polyhedron24_class::adjustcaptomatchvolume Interface Reference

Public Member Functions

- subroutine **polyhedron24_class_adjustcaptomatchvolume** (this, a_correct_signed_volume)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.14 f_polyhedron24_doubles3_class::adjustcaptomatchvolume Interface Reference

Public Member Functions

- subroutine **polyhedron24_doubles3_class_adjustcaptomatchvolume** (this, a_correct_signed_volume)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.15 IRL::AdvectionPlaneReconstruction Class Reference

Static Public Member Functions

- template<class MomentsContainerType , class CellType >
static [PlanarSeparator solve](#) (const MomentsContainerType &a_volume_moments_list, const R2PNeighborhood<CellType > &a_neighborhood, const double threshold=0.9)

Solve for the [AdvectionPlaneReconstruction](#), returning a [PlanarSeparator](#).

Static Private Member Functions

- template<class ContainedType , class CellType >
static **PlanarSeparator** **findBestPermutation** (ContainedType *a_moments, const **R2PNeighborhood**< CellType > &a_neighborhood, const double a_target_volume_fraction)
- template<class CellType >
static **PlanarSeparator** **constructSeparatorAttempt** (const CellType &a_cell, const double a_target_volume_fraction, const **Normal** &a_normal_0, const **Pt** &a_pt_0, const **Normal** &a_normal_1, const **Pt** &a_pt_1, const double a_flip_cut)
- template<class CellType >
static void **checkIfBest** (const **R2PNeighborhood**< CellType > &a_neighborhood, const **PlanarSeparator** &a_attempt_separator, **PlanarSeparator** *a_current_best_separator, double *a_current_minimum_error)

Friends

- class **AdvectionPlaneReconstructionDebug**

The documentation for this class was generated from the following file:

- advected_plane_reconstruction.h

6.16 IRL::AdvectionPlaneReconstructionDebug Class Reference

Static Public Member Functions

- template<class MomentsContainerType , class CellType >
static **PlanarSeparator** **solve** (const MomentsContainerType &a_volume_moments_list, const **R2PNeighborhood**< CellType > &a_neighborhood, const double threshold=0.9)

Static Private Member Functions

- template<class ContainedType , class CellType >
static **PlanarSeparator** **findBestPermutation** (ContainedType *a_moments, const **R2PNeighborhood**< CellType > &a_neighborhood, const double a_target_volume_fraction)
- template<class CellType >
static void **checkIfBest** (const **R2PNeighborhood**< CellType > &a_neighborhood, const **PlanarSeparator** &a_attempt_separator, **PlanarSeparator** *a_current_best_separator, double *a_current_minimum_error)
- template<class CellType >
static void **writeOutPlane** (const **R2PNeighborhood**< CellType > &a_neighborhood, const **PlanarSeparator** &a_reconstruction, const std::string &a_prefix, const std::size_t a_iteration_number)
- template<class CellType >
static void **writeOutCentroids** (const **R2PNeighborhood**< CellType > &a_neighborhood)

The documentation for this class was generated from the following file:

- advected_plane_reconstruction.h

6.17 f_listedvm_vman_class::append Interface Reference

Public Member Functions

- subroutine **listedvm_vman_class_append** (this, a_other_list)

The documentation for this interface was generated from the following file:

- [f_listedvm_vman_class.f90](#)

6.18 f_tagged_accumlistedvm_vman_class::append Interface Reference

Public Member Functions

- subroutine **tagged_accumlistedvm_vman_class_append** (this, a_other_list)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.19 short_alloc::arena< N, alignment > Class Template Reference

Public Member Functions

- **arena** (const [arena](#) &)=delete
- **arena** & **operator=** (const [arena](#) &)=delete
- template<std::size_t ReqAlign>
char * **allocate** (std::size_t n)
- void **deallocate** (char *p, std::size_t n) noexcept
- std::size_t **used** () const noexcept
- void **reset** () noexcept

Static Public Member Functions

- static constexpr std::size_t **size** () noexcept

Private Member Functions

- bool **pointer_in_buffer** (char *p) noexcept

Static Private Member Functions

- static std::size_t **align_up** (std::size_t n) noexcept

Private Attributes

- char **buf_**[N]
- char * **ptr_**

The documentation for this class was generated from the following file:

- short_alloc.h

6.20 IRL::BFGS< OptimizingClass, kParameters > Class Template Reference

Implementation of [BFGS](#) method according to the book Numerical Optimization by Jorge Nocedal and Stephen Wright, 2006.

```
#include <bfgs.h>
```

Public Member Functions

- void [solve](#) (OptimizingClass *a_setup_otype, const Eigen::Matrix< double, kParameters, 1 > &a_gradient_< _delta)

Assigns otype_m to a new pointer and then solves.
- int [getReason](#) (void)

Return reason for exiting by integer.
- UnsignedIndex_t [getIterationCount](#) (void)

Return the number of iterations it took until exit.
- [~BFGS](#) (void)=default

Default destructor.

Private Member Functions

- void [solve](#) (const Eigen::Matrix< double, kParameters, 1 > &a_gradient_delta)

Perform non-linear optimization.
- double [calculateCostFunction](#) (Eigen::Matrix< double, kParameters, 1 > *a_travel_amount)

Calculate cost function with a given travel amount. This travel amount is the displacement (in parameter space) from the last accepted ('bestGuess') solution. If this is assumed to be represented as f(x) in typical BFGS literature, a travel amount would then be p_k, or the step distance times the step direction.
- Eigen::Matrix< double, kParameters, 1 > [calculateGradientOfCostFunction](#) (const Eigen::Matrix< double, kParameters, 1 > &a_gradient_delta)

Calculate cost function gradient. Calculated at a_starting_delta via forward difference with the step taken for finite-difference determined by parameter's entry in a_gradient_delta.
- Eigen::Matrix< double, kParameters, 1 > [getStep](#) (const Eigen::Matrix< double, kParameters, 1 > &a_< search_direction)

Perform line search to determine step length.

Private Attributes

- int **reason_for_exit_m**
- UnsignedIndex_t **iteration_m**
- OptimizingClass * **otype_m**

Static Private Attributes

- static constexpr double **step_size_tolerance** = 1.0e-10

Friends

- class **StrongWolfeConditionLineSearch**

6.20.1 Detailed Description

```
template<class OptimizingClass, int kParameters>
class IRL::BFGS< OptimizingClass, kParameters >
```

Implementation of **BFGS** method according to the book Numerical Optimization by Jorge Nocedal and Stephen Wright, 2006.

Working implementation of **BFGS** non-linear optimization algorithm. The line search used is implemented below in the class **StrongWolfeConditionLineSearch**. This line search preserves the Strong Wolfe Conditions for step size and curvature, ensuring that the estimated Hessian remains Symmetric Positive Definite and a valid search direction can always be found.

Note: There is probably much opportunity for optimizing the implementation of the algorithm. Right now it is rather slow.

The documentation for this class was generated from the following file:

- bfgs.h

6.21 IRL::Bisection< OptimizingClass > Class Template Reference

A templated class driver that performs bisection.

```
#include <bisection.h>
```

Public Member Functions

- **Bisection** (void)
Default constructor.
- void **solve** (OptimizingClass *a_otype, double a_bracket_0, double a_bracket_1, const UnsignedIndex_t a_max_iter=100000000)
Solution that resets optimizing class and brackets, then solves.
- **~Bisection** (void)=default
Default destructor.

Private Member Functions

- void **solve** (double a_bracket_0, double a_bracket_1, const UnsignedIndex_t a_max_iter)
Given two locations that bracket a zero, find the zero.
- double **calculateError** (double *a_guess)
Set the guess and calculate the associated error.

Private Attributes

- OptimizingClass * `otype_m`

Pointer to object of class OptimizingClass that is being optimized.

6.21.1 Detailed Description

```
template<class OptimizingClass>
class IRL::Bisection< OptimizingClass >
```

A templated class driver that performs bisection.

This class is a general bisection solver that takes a pointer to a templated class that must implement certain methods (detailed below). Upon completion, the final solution should be stored in the pointed-to-class because of the use of `setGuess()` called in `this->calculateError()`.

Requirements for Optimizing class: `calculateSignedScalarError()` : A method to calculate a scalar error for the class with sign of (correct-guess).

- `setGuess(...)` : A method that takes the guess value and applies it to be used when `calculateSignedScalarError()` is called.
- `errorTooHigh(...)` : A method that takes error and returns a boolean whether the error is low enough to stop optimization and return.

6.21.2 Member Function Documentation

6.21.2.1 `solve()` [1/2]

```
template<class OptimizingClass >
void IRL::Bisection< OptimizingClass >::solve (
    OptimizingClass * a_otype,
    double a_bracket_0,
    double a_bracket_1,
    const UnsignedIndex_t a_max_iter = 100000000 )
```

Solution that resets optimizing class and brackets, then solves.

Parameters

in	<code>a_otype</code>	Pointer to class being optimized
in	<code>a_bracket_0</code>	Bracket on one side of zero.
in	<code>a_bracket_1</code>	Bracket on other side of zero from <code>a_bracket_0</code> .

6.21.2.2 solve() [2/2]

```
template<class OptimizingClass >
void IRL::Bisection< OptimizingClass >::solve (
    double a_bracket_0,
    double a_bracket_1,
    const UnsignedIndex_t a_max_iter )  [private]
```

Given two locations that bracket a zero, find the zero.

Parameters

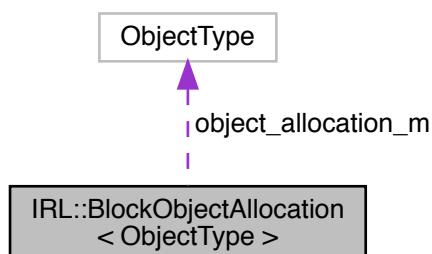
in	<i>a_bracket_0</i>	Bracket on one side of zero.
in	<i>a_bracket_1</i>	Bracket on other side of zero from a_bracket_0.

The documentation for this class was generated from the following file:

- bisection.h

6.22 IRL::BlockObjectAllocation< ObjectType > Class Template Reference

Collaboration diagram for IRL::BlockObjectAllocation< ObjectType >:



Public Member Functions

- **BlockObjectAllocation** (const LargeOffsetIndex_t a_number_to_allocate)
- ObjectType * **getObjectAtIndex** (const LargeOffsetIndex_t a_object_index)
- **BlockObjectAllocation** (const [BlockObjectAllocation](#) &other)
- **BlockObjectAllocation** ([BlockObjectAllocation](#) &&other) noexcept
- **BlockObjectAllocation** & **operator=** (const [BlockObjectAllocation](#) &other)
- **BlockObjectAllocation** & **operator=** ([BlockObjectAllocation](#) &&other) noexcept

Private Attributes

- `ObjectType * object_allocation_m`
- `LargeOffsetIndex_t allocated_size_m`

The documentation for this class was generated from the following file:

- `block_object_allocation.h`

6.23 IRL::BrentsMethod< OptimizingClass > Class Template Reference

A templated class driver that performs Brent's Method.

```
#include <brents_method.h>
```

Public Member Functions

- `BrentsMethod (void)`
Default constructor.
- `void solve (OptimizingClass *a_otype, double a_bracket_0, double a_bracket_1, const double a_delta)`
Solution that resets optimizing class and brackets, then solves.
- `~BrentsMethod (void)=default`
Default destructor.

Private Member Functions

- `void solve (double a_bracket_0, double a_bracket_1, const double a_delta)`
Given two locations that bracket a zero, find the zero.
- `double calculateError (double *a_guess)`
Set the guess and calculate the associated error.

Private Attributes

- `OptimizingClass * otype_m`
Pointer to object of class `OptimizingClass` that is being optimized.

6.23.1 Detailed Description

```
template<class OptimizingClass>
class IRL::BrentsMethod< OptimizingClass >
```

A templated class driver that performs Brent's Method.

This class is a general Brent's Method solver that takes a pointer to a templated class that must implement certain methods (detailed below). Upon completion, the final solution should be stored in the pointed-to-class because of the use of `setGuess ()` before exiting `this->solve ()`.

Requirements for Optimizing class: `calculateSignedScalarError ()` : A method to calculate a scalar error for the class with sign of (correct-guess).

- `setGuess (...)` : A method that takes the guess value and applies it to be used when `calculate← SignedScalarError ()` is called.
- `errorTooHigh (...)` : A method that takes error and returns a boolean whether the error is low enough to stop optimization and return.

6.23.2 Member Function Documentation

6.23.2.1 solve() [1/2]

```
template<class OptimizingClass >
void IRL::BrentsMethod< OptimizingClass >::solve (
    OptimizingClass * a_otype,
    double a_bracket_0,
    double a_bracket_1,
    const double a_delta )
```

Solution that resets optimizing class and brackets, then solves.

Parameters

in	<i>a_otype</i>	Pointer to class being optimized
in	<i>a_bracket_0</i>	Bracket on one side of zero.
in	<i>a_bracket_1</i>	Bracket on other side of zero from <i>a_bracket_0</i> .
in	<i>a_delta</i>	Value for delta that dictates which method is used inside Brents Method.

6.23.2.2 solve() [2/2]

```
template<class OptimizingClass >
void IRL::BrentsMethod< OptimizingClass >::solve (
    double a_bracket_0,
    double a_bracket_1,
    const double a_delta ) [private]
```

Given two locations that bracket a zero, find the zero.

Parameters

in	<i>a_bracket_0</i>	Bracket on one side of zero.
in	<i>a_bracket_1</i>	Bracket on other side of zero from <i>a_bracket_0</i> .
in	<i>a_delta</i>	Value for delta that dictates which method is used inside Brents Method.

The documentation for this class was generated from the following file:

- brents_method.h

6.24 IRL::ByteBuffer Class Reference

Public Member Functions

- `ByteBuffer (void)`
Default constructor.
- `LargeOffsetIndex_t size (void) const`
Return const size of the buffer.
- `void resize (const LargeOffsetIndex_t a_index)`
Resize the underlying buffer array.
- `Byte_t * data (void)`
Return object pointer to at start of buffer_m.
- `const Byte_t * data (void) const`
Return const object pointer to at start of buffer_m.
- `void resetBufferPointer (void)`
Resets the buffer location index to 0 (the start).
- `void reserve (const LargeOffsetIndex_t a_number_to_reserve)`
Allow reservation of space to gain better performance if size is known a priori.
- template<class ObjectType >
`void pack (const ObjectType *a_list, const LargeOffsetIndex_t a_number_of_elements)`
Pack object (or list of objects) onto the end of the buffer.
- template<class ObjectType >
`void unpack (ObjectType *a_list, const LargeOffsetIndex_t a_number_of_elements)`
Unpack bytes in list, starting at current buffer location into objects pointed to by a_list.
- `~ByteBuffer (void)=default`
Default destructor.

Private Member Functions

- `Byte_t * getBufferLocationPointer (void)`
Get pointer to underlying data at current buffer location.
- `Byte_t * getOffsetBufferLocationPointer (const LargeOffsetIndex_t a_offset)`
Get pointer to underlying data at offset from current buffer location.
- `void setBufferLocation (const LargeOffsetIndex_t a_index)`
Set buffer location index to a_index.
- `void advanceBufferLocation (const LargeOffsetIndex_t a_increment)`
Update buffer location.

Private Attributes

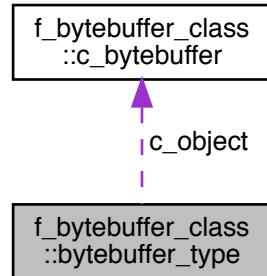
- `std::vector< Byte_t > buffer_m`
Vector of bytes.
- `LargeOffsetIndex_t buffer_location_m`
Current buffer location.

The documentation for this class was generated from the following file:

- `byte_buffer.h`

6.25 f_bytebuffer_class::bytebuffer_type Type Reference

Collaboration diagram for f_bytebuffer_class::bytebuffer_type:



Public Member Functions

- final **bytebuffer_class_delete**

Private Attributes

- type([c_bytebuffer](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.26 f_bytebuffer_class::c_bytebuffer Type Reference

Private Attributes

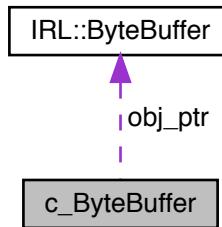
- type([c_ptr](#)), private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.27 c_ByteBuffer Struct Reference

Collaboration diagram for c_ByteBuffer:



Public Attributes

- `IRL::ByteBuffer * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_byte_buffer.h`

6.28 c_CappedDodecahedron Struct Reference

Collaboration diagram for c_CappedDodecahedron:



Public Attributes

- `IRL::CappedDodecahedron * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_capped_dodecahedron.h`

6.29 f_cappeddodecahedron_class::c_cappeddodecahedron Type Reference

Private Attributes

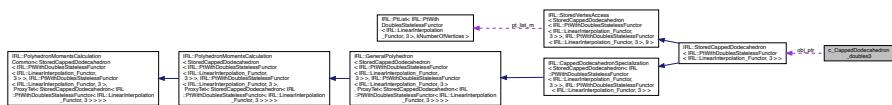
- `type(c_ptr)`, private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- f_cappeddodecahedron_class.f90

6.30 c_CappedDodecahedron_doubles3 Struct Reference

Collaboration diagram for c_CappedDodecahedron_doubles3:



Public Attributes

- `IRL::StoredCappedDodecahedron < IRL::PtWithDoublesStatelessFunctor < IRL::LinearInterpolation_Functor`,
`3 >> * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- c capped dodecahedron doubles3.h

6.31 f_cappeddodecahedron_doubles3_class::c_cappeddodecahedron_doubles3 Type Reference

Private Attributes

- type(c_ptr), private **object** = C NULL_PTR

The documentation for this type was generated from the following file:

- f_cappeddodecahedron doubles3 class.f90

6.32 c_DividedPolygon Struct Reference

Collaboration diagram for c_DividedPolygon:



Public Attributes

- `IRL::DividedPolygon * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_divided_polygon.h`

6.33 f_dividedpolygon_class::c_dividedpolygon Type Reference

Private Attributes

- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_dividedpolygon_class.f90`

6.34 f_dodecahedron_class::c_dodecahedron Type Reference

Private Attributes

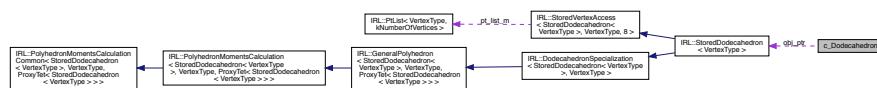
- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_dodecahedron_class.f90`

6.35 c_Dodecahedron Struct Reference

Collaboration diagram for c_Dodecahedron:



Public Attributes

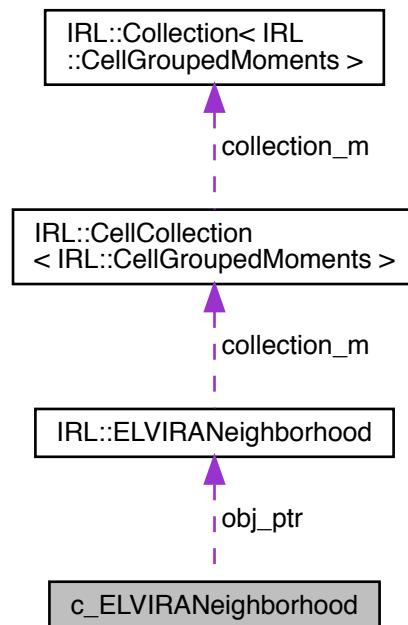
- `IRL::Dodecahedron * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_dodecahedron.h`

6.36 c_ELVIRANeighborhood Struct Reference

Collaboration diagram for `c_ELVIRANeighborhood`:



Public Attributes

- `IRL::ELVIRANeighborhood * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_elvira_neighborhood.h`

6.37 f_elviraneighborhood_class::c_elviraneighborhood Type Reference

Private Attributes

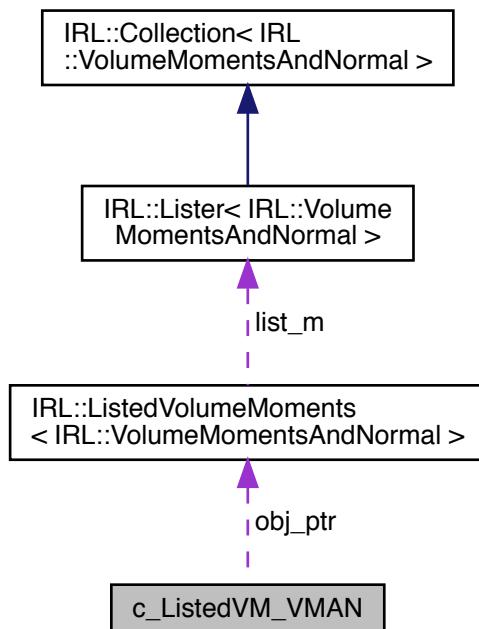
- type(c_ptr), private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- f_elviraneighborhood_class.f90

6.38 c_ListedVM_VMAN Struct Reference

Collaboration diagram for c_ListedVM_VMAN:



Public Attributes

- `IRL::ListedVolumeMoments<IRL::VolumeMomentsAndNormal> * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- c_listedvm_vman.h

6.39 f_listedvm_vman_class::c_listedvm_vman Type Reference

Private Attributes

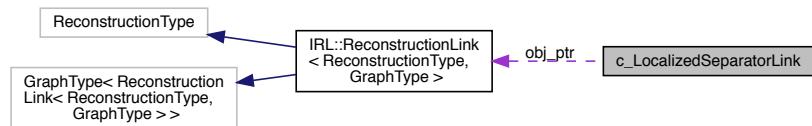
- type(c_ptr), private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- f_listedvm_vman_class.f90

6.40 c_LocalizedSeparatorLink Struct Reference

Collaboration diagram for c_LocalizedSeparatorLink:



Public Attributes

- IRL::LocalizedSeparatorLink * **obj_ptr**
- bool **allocated_from_object_allocation_server** = false

The documentation for this struct was generated from the following file:

- c_localized_separator_link.h

6.41 f_localizedseparatorlink_class::c_localizedseparatorlink Type Reference

Private Attributes

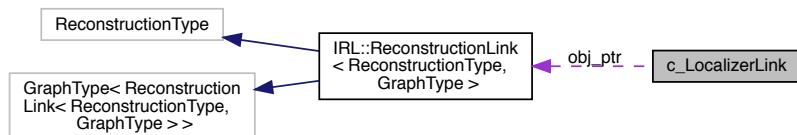
- type(c_ptr), private **object** = C_NULL_PTR
- logical(c_bool), private **allocated_from_object_allocation_server** = .false.

The documentation for this type was generated from the following file:

- f_localizedseparatorlink_class.f90

6.42 c_LocalizerLink Struct Reference

Collaboration diagram for c_LocalizerLink:



Public Attributes

- `IRL::LocalizerLink * obj_ptr = nullptr`
- `bool allocated_from_object_allocation_server = false`

The documentation for this struct was generated from the following file:

- `c_localizer_link.h`

6.43 f_localizerlink_class::c_localizerlink Type Reference

Private Attributes

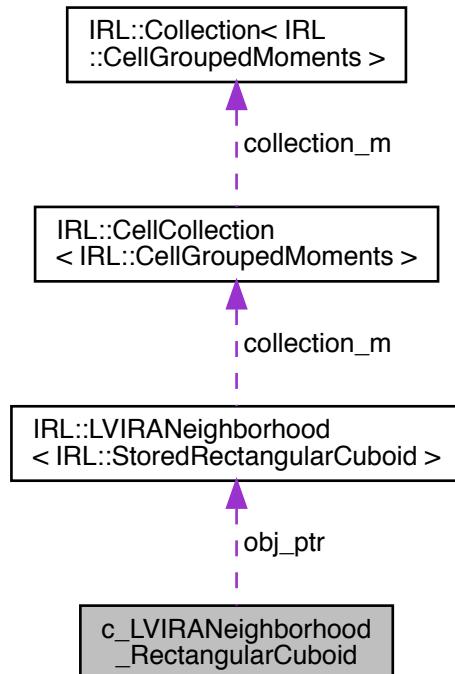
- `type(c_ptr), private object = C_NULL_PTR`
- `logical(c_bool), private allocated_from_object_allocation_server = .false.`

The documentation for this type was generated from the following file:

- `f_localizerlink_class.f90`

6.44 c_LVIRANeighborhood_RectangularCuboid Struct Reference

Collaboration diagram for c_LVIRANeighborhood_RectangularCuboid:



Public Attributes

- `IRL::LVIRANeighborhood< IRL::RectangularCuboid > * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_lvira_neighborhood_rectangular_cuboid.h`

6.45 f_lviraneighborhood_rectangularcuboid_class::c_lviraneighborhood_rectangularcuboid Type Reference

Private Attributes

- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_lviraneighborhood_rectangularcuboid_class.f90`

6.46 c_ObjectAllocationServer_LocalizedSeparatorLink Struct Reference

Collaboration diagram for c_ObjectAllocationServer_LocalizedSeparatorLink:



Public Attributes

- `IRL::ObjectAllocationServer< IRL::LocalizedSeparatorLink > * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_object_allocation_server_localized_separator_link.h`

6.47 f_objectallocationserver_localizedseparatorlink_class::c_objectallocationserver_localizedseparatorlink Type Reference

Private Attributes

- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_objectallocationserver_localizedseparatorlink_class.f90`

6.48 f_objectallocationserver_localizerlink_class::c_objectallocationserver_localizerlink Type Reference

Private Attributes

- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_objectallocationserver_localizerlink_class.f90`

6.49 c_ObjectAllocationServer_LocalizerLink Struct Reference

Collaboration diagram for c_ObjectAllocationServer_LocalizerLink:



Public Attributes

- `IRL::ObjectAllocationServer< IRL::LocalizerLink > * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_object_allocation_server_localizer_link.h`

6.50 f_objectallocationserver_planarlocalizer_class::c_objectallocationserver_planarlocalizer Type Reference

Private Attributes

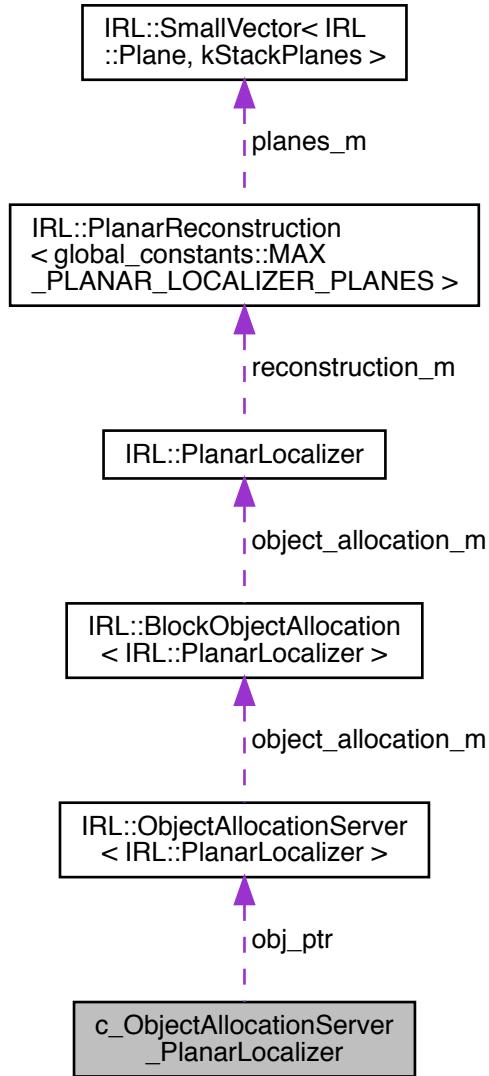
- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_objectallocationserver_planarlocalizer_class.f90`

6.51 c_ObjectAllocationServer_PlanarLocalizer Struct Reference

Collaboration diagram for c_ObjectAllocationServer_PlanarLocalizer:



Public Attributes

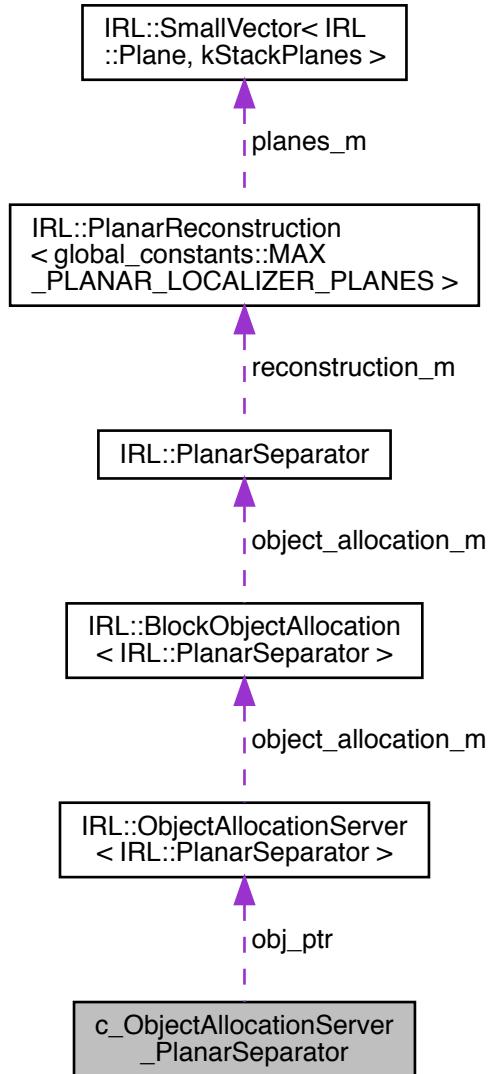
- `IRL::ObjectAllocationServer< IRL::PlanarLocalizer > * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_object_allocation_server_planar_localizer.h`

6.52 c_ObjectAllocationServer_PlanarSeparator Struct Reference

Collaboration diagram for c_ObjectAllocationServer_PlanarSeparator:



Public Attributes

- `IRL::ObjectAllocationServer< IRL::PlanarSeparator > * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_object_allocation_server_planar_separator.h`

6.53 f_objectallocationserver_planarseparator_class::c_objectallocationserver_planarseparator Type Reference

Private Attributes

- type(c_ptr), private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- [f_objectallocationserver_planarseparator_class.f90](#)

6.54 f_planarlocalizer_class::c_planarlocalizer Type Reference

Private Attributes

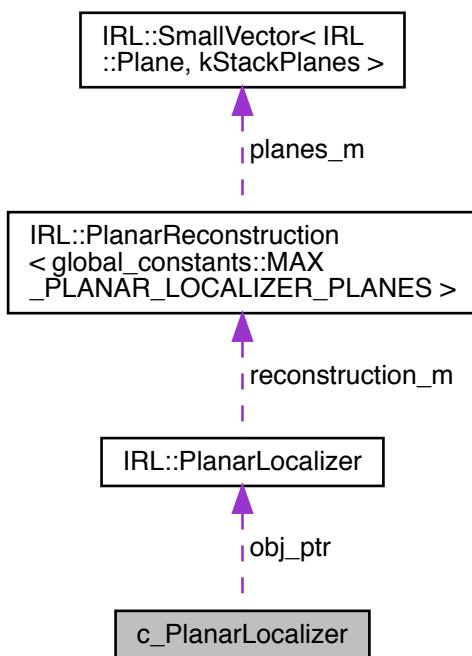
- type(c_ptr), private **object** = C_NULL_PTR
- logical(c_bool), private **allocated_from_object_allocation_server** = .false.

The documentation for this type was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.55 c_PlanarLocalizer Struct Reference

Collaboration diagram for c_PlanarLocalizer:



Public Attributes

- `IRL::PlanarLocalizer * obj_ptr = nullptr`
- `bool allocated_from_object_allocation_server = false`

The documentation for this struct was generated from the following file:

- [c_localizers.h](#)

6.56 f_planarseparator_class::c_planarseparator Type Reference

Private Attributes

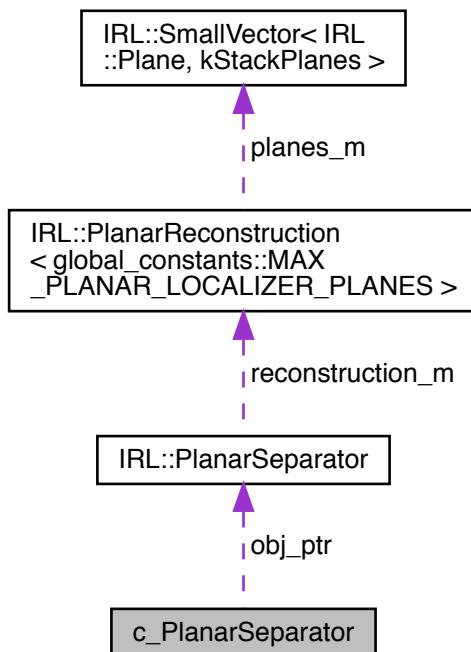
- `type(c_ptr), private object = C_NULL_PTR`
- `logical(c_bool), private allocated_from_object_allocation_server = .false.`

The documentation for this type was generated from the following file:

- [f_planarseparator_class.f90](#)

6.57 c_PlanarSeparator Struct Reference

Collaboration diagram for c_PlanarSeparator:



Public Attributes

- `IRL::PlanarSeparator * obj_ptr = nullptr`
- `bool allocated_from_object_allocation_server = false`

The documentation for this struct was generated from the following file:

- `c_separators.h`

6.58 f_polygon_class::c_polygon Type Reference

Private Attributes

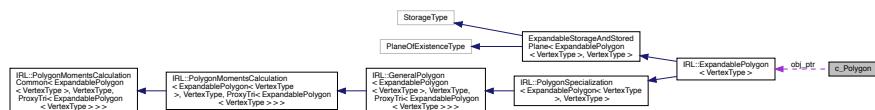
- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_polygon_class.f90`

6.59 c_Polygon Struct Reference

Collaboration diagram for `c_Polygon`:



Public Attributes

- `IRL::Polygon * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_polygon.h`

6.60 f_polyhedron24_class::c_polyhedron24 Type Reference

Private Attributes

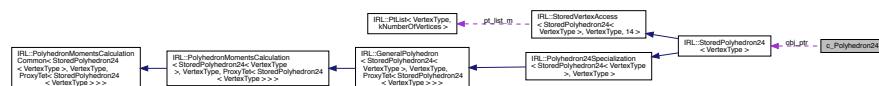
- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_polyhedron24_class.f90`

6.61 c_Polyhedron24 Struct Reference

Collaboration diagram for c_Polyhedron24:



Public Attributes

- IRL::Polyhedron24 * **obj_ptr** = nullptr

The documentation for this struct was generated from the following file:

- c_polyhedron24.h

6.62 f_polyhedron24_doubles3_class::c_polyhedron24_doubles3 Type Reference

Private Attributes

- `type(c_ptr)`, private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- f_polyhedron24_doubles3_class.f90

6.63 c_Polyhedron24_doubles3 Struct Reference

Collaboration diagram for c_Polyhedron24_doubles3:



Public Attributes

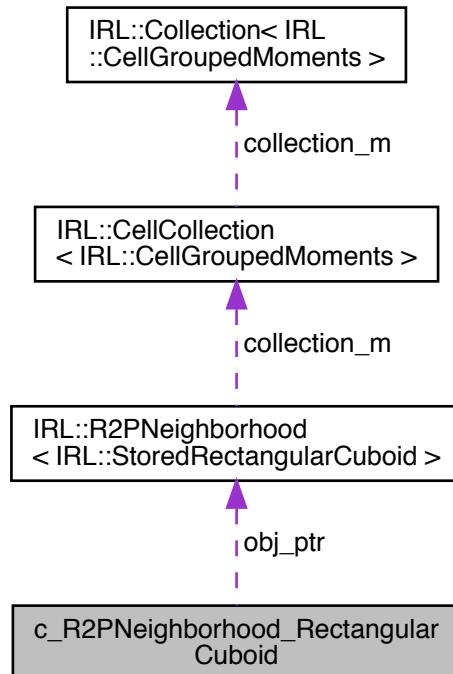
- `IRL::StoredPolyhedron24< IRL::PtWithDoublesStatelessFunctor< IRL::LinearInterpolation_Functor, 3 > >`
`* obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- c polyhedron24 doubles3.h

6.64 c_R2PNeighborhood_RectangularCuboid Struct Reference

Collaboration diagram for c_R2PNeighborhood_RectangularCuboid:



Public Attributes

- `IRL::R2PNeighborhood< IRL::RectangularCuboid > * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_r2p_neighborhood_rectangular_cuboid.h`

6.65 f_r2pneighborhood_rectangularcuboid_class::c_r2pneighborhood_rectangularcuboid Type Reference

Private Attributes

- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_r2pneighborhood_rectangularcuboid_class.f90`

6.66 c_RectangularCuboid Struct Reference

Collaboration diagram for c_RectangularCuboid:



Public Attributes

- `IRL::RectangularCuboid * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_rectangular_cuboid.h`

6.67 f_rectangularcuboid_class::c_rectangularcuboid Type Reference

Private Attributes

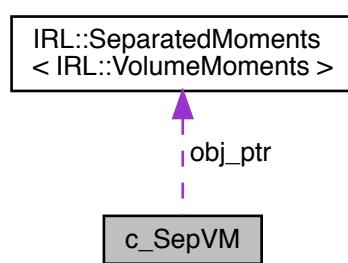
- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_rectangularcuboid_class.f90`

6.68 c_SepVM Struct Reference

Collaboration diagram for c_SepVM:



Public Attributes

- `IRL::SeparatedMoments< IRL::VolumeMoments > * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_separated_volume_moments.h`

6.69 f_sepvm_class::c_sepvm Type Reference

Private Attributes

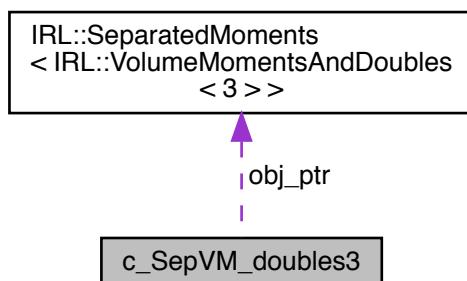
- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_sepvm_class.f90`

6.70 c_SepVM_doubles3 Struct Reference

Collaboration diagram for `c_SepVM_doubles3`:



Public Attributes

- `IRL::SeparatedMoments< IRL::VolumeMomentsAndDoubles< 3 > > * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_separated_volume_moments_doubles3.h`

6.71 f_sepvm_doubles3_class::c_sepvm_doubles3 Type Reference

Private Attributes

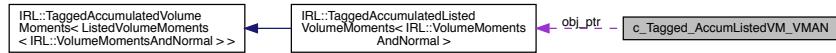
- type(c_ptr), private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.72 c_Tagged_AccumListedVM_VMAN Struct Reference

Collaboration diagram for c_Tagged_AccumListedVM_VMAN:



Public Attributes

- [IRL::TaggedAccumulatedListedVolumeMoments< IRL::VolumeMomentsAndNormal > * obj_ptr = nullptr](#)

The documentation for this struct was generated from the following file:

- [c_tagged_accumulated_listed_volume_moments_and_normal.h](#)

6.73 f_tagged_accumlistedvm_vman_class::c_tagged_accumlistedvm_vman Type Reference

Private Attributes

- type(c_ptr), private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.74 f_tagged_accumvm_sepvm_class::c_tagged_accumvm_sepvm Type Reference

Private Attributes

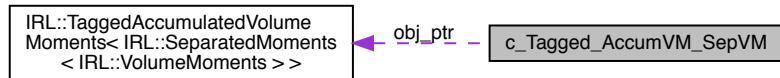
- type(c_ptr), private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.75 c_Tagged_AccumVM_SepVM Struct Reference

Collaboration diagram for c_Tagged_AccumVM_SepVM:



Public Attributes

- [IRL::TaggedAccumulatedVolumeMoments< IRL::SeparatedMoments< IRL::VolumeMoments >> * obj_ptr = nullptr](#)

The documentation for this struct was generated from the following file:

- [c_tagged_accumulated_separated_volume_moments.h](#)

6.76 f_tagged_accumvm_vm_class::c_tagged_accumvm_vm Type Reference

Private Attributes

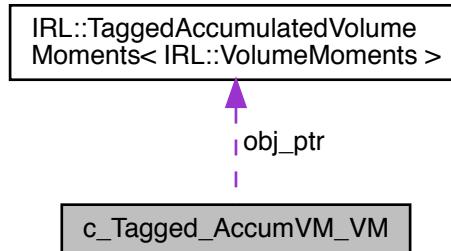
- type(c_ptr), private **object** = C_NULL_PTR

The documentation for this type was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.77 c_Tagged_AccumVM_VM Struct Reference

Collaboration diagram for c_Tagged_AccumVM_VM:



Public Attributes

- `IRL::TaggedAccumulatedVolumeMoments< IRL::VolumeMoments > * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_tagged_accumulated_volume_moments.h`

6.78 f_tet_class::c_tet Type Reference

Private Attributes

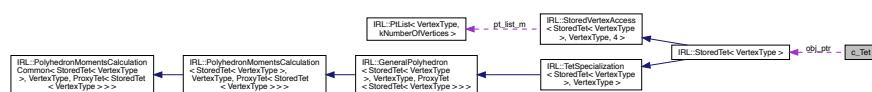
- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_tet_class.f90`

6.79 c_Tet Struct Reference

Collaboration diagram for c_Tet:



Public Attributes

- `IRL::Tet * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_tet.h`

6.80 f_tri_class::c_tri Type Reference

Private Attributes

- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_tri_class.f90`

6.81 c_Tri Struct Reference

Collaboration diagram for `c_Tri`:



Public Attributes

- `IRL::Tri * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_tri.h`

6.82 f_vman_class::c_vman Type Reference

Private Attributes

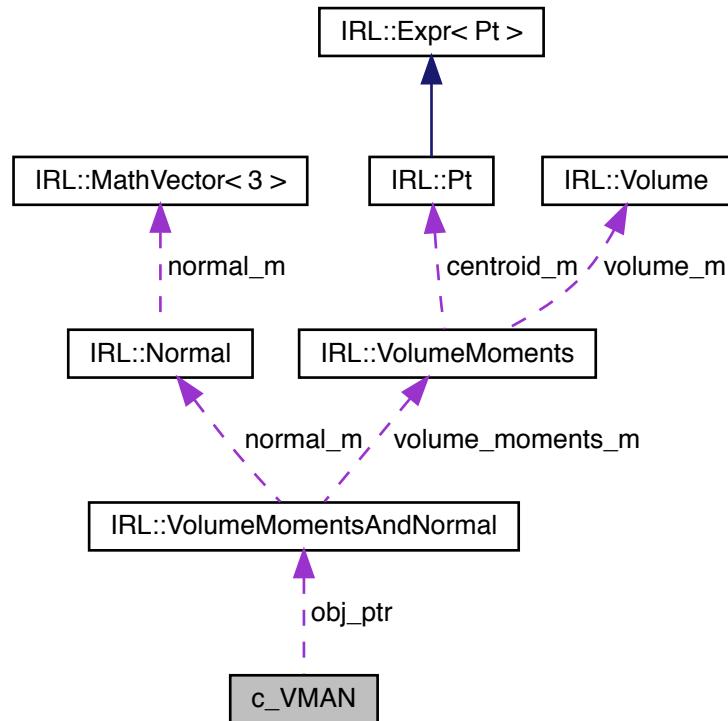
- `type(c_ptr), private object = C_NULL_PTR`

The documentation for this type was generated from the following file:

- `f_vman_class.f90`

6.83 c_VMAN Struct Reference

Collaboration diagram for c_VMAN:



Public Attributes

- `IRL::VolumeMomentsAndNormal * obj_ptr = nullptr`

The documentation for this struct was generated from the following file:

- `c_volume_moments_and_normal.h`

6.84 f_tri_class::calculateandsetplaneofexistence Interface Reference

Public Member Functions

- subroutine `tri_class_calculateandsetplaneofexistence` (this)

The documentation for this interface was generated from the following file:

- `f_tri_class.f90`

6.85 f_polygon_class::calculateandsetplaneofexistence Interface Reference

Public Member Functions

- subroutine **polygon_class_calculateandsetplaneofexistence** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.86 f_dividedpolygon_class::calculateandsetplaneofexistence Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_calculateandsetplaneofexistence** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.87 f_tri_class::calculatecentroid Interface Reference

Public Member Functions

- real(rl_double) function, dimension(1:3) **tri_class_calculatecentroid** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.88 f_polygon_class::calculatecentroid Interface Reference

Public Member Functions

- real(rl_double) function, dimension(3) **polygon_class_calculatecentroid** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.89 f_polygon_class::calculatenarestptonsurface Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **polygon_class_calculatenarestptonsurface** (this, a_pt)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.90 f_tri_class::calculatenormal Interface Reference

Public Member Functions

- real(irl_double) function, dimension(1:3) **tri_class_calculatenormal** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.91 f_polygon_class::calculatenormal Interface Reference

Public Member Functions

- real(irl_double) function, dimension(1:3) **polygon_class_calculatenormal** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.92 f_dividedpolygon_class::calculatenormal Interface Reference

Public Member Functions

- real(irl_double) function, dimension(1:3) **dividedpolygon_class_calculatenormal** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.93 f_tri_class::calculatesign Interface Reference

Public Member Functions

- real(irl_double) function **tri_class_calculatesign** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.94 f_polygon_class::calculatesign Interface Reference

Public Member Functions

- real(irl_double) function **polygon_class_calculatesign** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.95 f_dividedpolygon_class::calculatesign Interface Reference

Public Member Functions

- real(irl_double) function **dividedpolygon_class_calculatesign** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.96 f_dividedpolygon_class::calculatesurfacearea Interface Reference

Public Member Functions

- real(irl_double) function **dividedpolygon_class_calculatesurfacearea** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.97 f_tri_class::calculatevolume Interface Reference

Public Member Functions

- real(irl_double) function **tri_class_calculatevolume** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.98 f_polygon_class::calculatevolume Interface Reference

Public Member Functions

- real(irl_double) function **polygon_class_calculatevolume** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.99 f_rectangularcuboid_class::calculatevolume Interface Reference

Public Member Functions

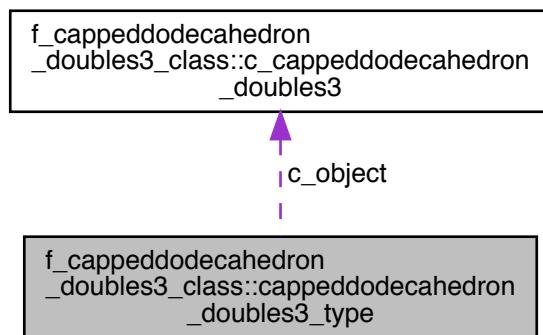
- real(irl_double) function **rectangularcuboid_class_calculatevolume** (this)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.100 f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_type Type Reference

Collaboration diagram for f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_type:



Public Member Functions

- final **cappeddodecahedron_doubles3_class_delete**

Private Attributes

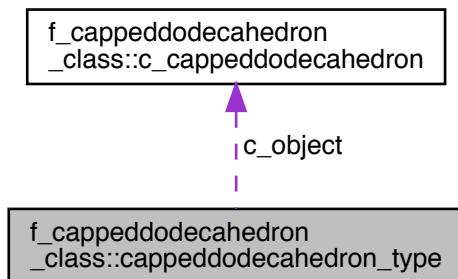
- type([c_cappeddodecahedron_doubles3](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.101 f_cappeddodecahedron_class::cappeddodecahedron_type Type Reference

Collaboration diagram for f_cappeddodecahedron_class::cappeddodecahedron_type:



Public Member Functions

- final **cappeddodecahedron_class_delete**

Private Attributes

- type([c_cappeddodecahedron](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.102 IRL::CappedDodecahedronSpecialization< Derived, VertexType > Class Template Reference

Inheritance diagram for IRL::CappedDodecahedronSpecialization< Derived, VertexType >:



Collaboration diagram for IRL::CappedDodecahedronSpecialization< Derived, VertexType >:



Public Member Functions

- [HalfEdgePolyhedron< VertexType > generateHalfEdgeVersion](#) (void) const
- template<class HalfEdgePolyhedronType >
[void setHalfEdgeVersion](#) (HalfEdgePolyhedronType *a_half_edge_version) const
- [ProxyTet< Derived > getSimplexFromDecomposition](#) (const UnsignedIndex_t a_tet) const
- [void adjustCapToMatchVolume](#) (const [Volume](#) a_correct_volume)

Static Public Member Functions

- static constexpr UnsignedIndex_t [getNumberOfSimplicesInDecomposition](#) (void)
- static constexpr std::array< UnsignedIndex_t, 4 > [getSimplexIndicesFromDecomposition](#) (const UnsignedIndex_t a_tet)

Additional Inherited Members

The documentation for this class was generated from the following file:

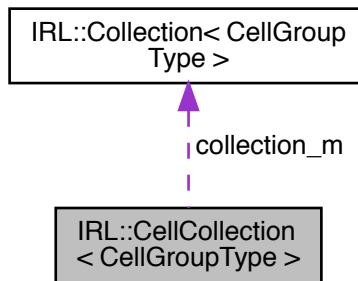
- [capped_dodecahedron.h](#)

6.103 IRL::CellCollection< CellGroupType > Class Template Reference

Class that contains a list of cells and coupled to moments.

```
#include <cell_collection.h>
```

Collaboration diagram for IRL::CellCollection< CellGroupType >:



Public Types

- using **iterator** = typename [Collection< CellGroupType >::iterator](#)
- using **const_iterator** = typename [Collection< CellGroupType >::const_iterator](#)
- using **contained_type** = [CellGroupType](#)

Public Member Functions

- [CellCollection \(void\)=default](#)
Default constructor.
- const [CellGroupType::cell_type & getCell \(const UnsignedIndex_t a_index\) const](#)
- const [CellGroupType::contained_type & getStoredMoments \(const UnsignedIndex_t a_index\) const](#)
- UnsignedIndex_t [size \(void\) const](#)
Get size of the collection.
- void [resize \(const UnsignedIndex_t a_size\)](#)
Set number of cells that are going to be involved in the collection.
- void [push_back \(const CellGroupType &a_object\)](#)
Add member to end of collection vector.
- void [clear \(void\)](#)
Empty the container.
- CellGroupType & [operator\[\] \(const UnsignedIndex_t a_index\)](#)
Get a member from the collection.
- const CellGroupType & [operator\[\] \(const UnsignedIndex_t a_index\) const](#)
Const version of getMember.
- iterator [begin \(void\) noexcept](#)
- const_iterator [begin \(void\) const noexcept](#)
- const_iterator [cbegin \(void\) const noexcept](#)
- iterator [end \(void\) noexcept](#)
- const_iterator [end \(void\) const noexcept](#)
- const_iterator [cend \(void\) const noexcept](#)
- [~CellCollection \(void\)=default](#)
Default destructor.

Private Member Functions

- void **checkIndex** (const UnsignedIndex_t a_index) const

Private Attributes

- **Collection< CellGroupType >** `collection_m`
Collection of CellGroupedMoments.

6.103.1 Detailed Description

```
template<class CellGroupType>
class IRL::CellCollection< CellGroupType >
```

Class that contains a list of cells and coupled to moments.

The documentation for this class was generated from the following file:

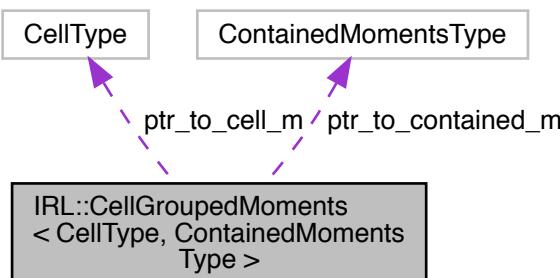
- `cell_collection.h`

6.104 IRL::CellGroupedMoments< CellType, ContainedMomentsType > Class Template Reference

A class that couples together a cell and another class.

```
#include <cell_grouped_moments.h>
```

Collaboration diagram for IRL::CellGroupedMoments< CellType, ContainedMomentsType >:



Public Types

- using **cell_type** = CellType
- using **contained_type** = ContainedMomentsType

Public Member Functions

- [CellGroupedMoments](#) (void)
Default constructor.
- [CellGroupedMoments](#) (const CellType *a_cell, const ContainedMomentsType *a_contained_object)
Construct given pointers to a_cell and a_contained_type.
- const CellType & [getCell](#) (void) const
Return const reference to the pointed to cell.
- const ContainedMomentsType & [getStoredMoments](#) (void) const
Return const reference to the pointed to cell.
- ContainedMomentsType [calculateNormalizedVolumeMoments](#) (const PlanarSeparator &a_separator) const
Return the normalized ContainedType for the cell given a_separator.
- [~CellGroupedMoments](#) (void)=default
Default destructor.

Private Attributes

- const CellType * **ptr_to_cell_m**
- const ContainedMomentsType * **ptr_to_contained_m**

6.104.1 Detailed Description

```
template<class CellType, class ContainedMomentsType>
class IRL::CellGroupedMoments< CellType, ContainedMomentsType >
```

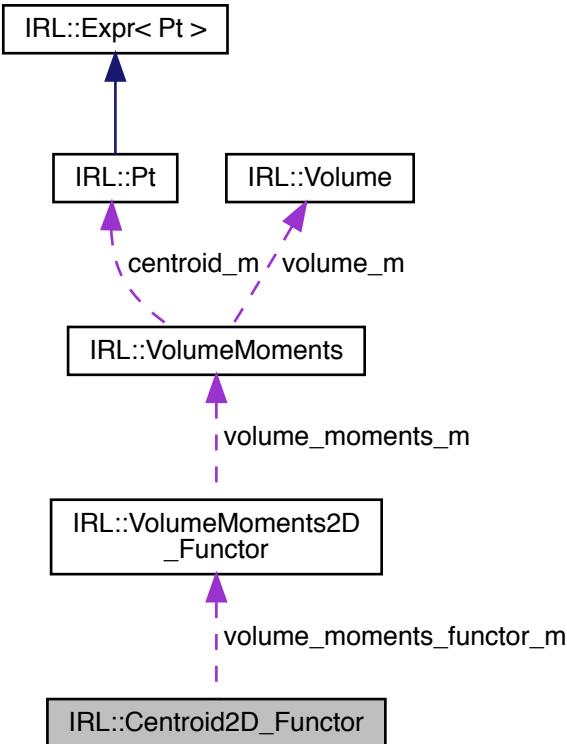
A class that couples together a cell and another class.

The documentation for this class was generated from the following file:

- cell_grouped_moments.h

6.105 IRL::Centroid2D_Functor Class Reference

Collaboration diagram for IRL::Centroid2D_Functor:



Public Types

- using **ReturnType** = **Pt**

Public Member Functions

- template<class SimplexType>
void **operator()** (const SimplexType &a_simplex)
- **ReturnType** **getMoments** (void) const

Private Attributes

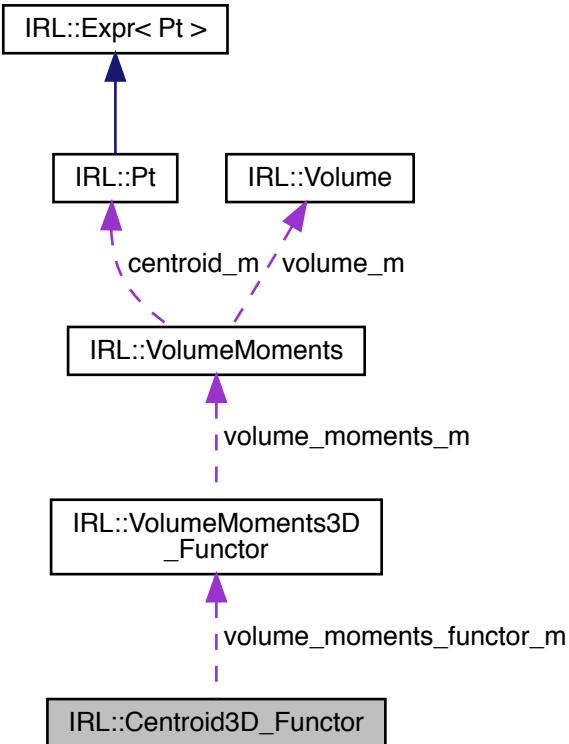
- **VolumeMoments2D_Functor** `volume_moments_functor_m`

The documentation for this class was generated from the following file:

- moment_calculation_through_simplices.h

6.106 IRL::Centroid3D_Functor Class Reference

Collaboration diagram for IRL::Centroid3D_Functor:



Public Types

- using **ReturnType** = **Pt**

Public Member Functions

- template<class SimplexType>
void **operator()** (const SimplexType &a_simplex)
- **ReturnType** **getMoments** (void) const

Private Attributes

- **VolumeMoments3D_Functor** **volume_moments_functor_m**

The documentation for this class was generated from the following file:

- moment_calculation_through_simplices.h

6.107 IRL::ChainedBlockStorage< ContainedType, kBlockSize > Class Template Reference

Public Types

- using **iterator** = [IteratorThroughBracketOperator< ChainedBlockStorage< ContainedType, kBlockSize > >](#)
- using **const_iterator** = [ConstIteratorThroughBracketOperator< ChainedBlockStorage< ContainedType, kBlockSize > >](#)
- using **value_t** = ContainedType

Public Member Functions

- **ChainedBlockStorage** (const UnsignedIndex_t a_initial_block_number)
- ContainedType & **operator[]** (const UnsignedIndex_t a_index)
- const ContainedType & **operator[]** (const UnsignedIndex_t a_index) const
- ContainedType & **getNextElement** (void)
- void **push_back** (const ContainedType &a_object)
- void **emplace_back** (ContainedType &&a_object)
- void **pop_back** (void)
- ContainedType & **back** (void)
- const ContainedType & **back** (void) const
- bool **empty** (void) const
- UnsignedIndex_t **size** (void) const
- void **resize** (const UnsignedIndex_t a_size)
- UnsignedIndex_t **currentSupportedSize** (void) const
- void **reserve** (const UnsignedIndex_t a_size)
- void **deallocateMemory** (void)
- **ChainedBlockStorage** (const [ChainedBlockStorage](#) &a_rhs)
- **ChainedBlockStorage** & **operator=** (const [ChainedBlockStorage](#) &a_rhs)
- iterator **begin** (void)
- iterator **end** (void)
- const_iterator **begin** (void) const
- const_iterator **end** (void) const
- const_iterator **cbegin** (void) const
- const_iterator **cend** (void) const

Private Member Functions

- ContainedType * **allocateBlock** (void)
- void **increaseToSize** (const UnsignedIndex_t a_size)
- void **addBlockToFreeList** (const UnsignedIndex_t a_block_index)

Private Attributes

- std::vector< ContainedType * > **data_blocks_m**
- std::vector< ContainedType * > **free_list_m**

The documentation for this class was generated from the following file:

- [chained_block_storage.h](#)

6.108 f_listedvm_vman_class::clear Interface Reference

Public Member Functions

- subroutine **listedvm_vman_class_clear** (this)

The documentation for this interface was generated from the following file:

- [f_listedvm_vman_class.f90](#)

6.109 f_tagged_accumlistedvm_vman_class::clear Interface Reference

Public Member Functions

- subroutine **tagged_accumlistedvm_vman_class_clear** (this)

The documentation for this interface was generated from the following file:

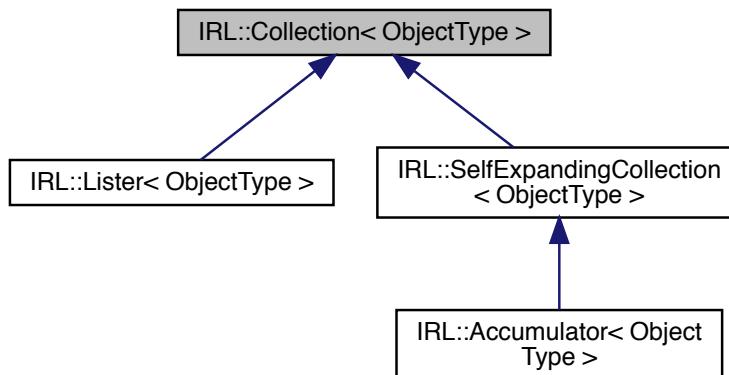
- [f_tagged_accumlistedvm_vman_class.f90](#)

6.110 IRL::Collection< ObjectType > Class Template Reference

Essentially a std::vector with assertions on access through operator[] in debug mode.

```
#include <collection.h>
```

Inheritance diagram for IRL::Collection< ObjectType >:



Public Types

- using **iterator** = typename std::vector< ObjectType >::iterator
- using **const_iterator** = typename std::vector< ObjectType >::const_iterator
- using **contained_type** = ObjectType

Public Member Functions

- **Collection** (void)=default
Default constructor.
- UnsignedIndex_t **size** (void) const
Get size of the collection.
- void **resize** (const UnsignedIndex_t a_size)
Set number of cells that are going to be involved in the collection.
- void **push_back** (const ObjectType &a_object)
Add member to end of collection vector.
- void **emplace_back** (const ObjectType &a_object)
Construct member at end of collection vector.
- void **clear** (void)
Empty the container.
- void **erase** (const UnsignedIndex_t a_index)
Erase an object from the container at the index.
- ObjectType & **operator[]** (const UnsignedIndex_t a_index)
Get a member from the collection.
- const ObjectType & **operator[]** (const UnsignedIndex_t a_index) const
Const version of getMember.
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept
- **~Collection** (void)=default
Default destructor.

Protected Member Functions

- void **checkIndex** (const UnsignedIndex_t a_index) const

Private Attributes

- std::vector< ObjectType > **collection_m**

6.110.1 Detailed Description

```
template<class ObjectType>
class IRL::Collection< ObjectType >
```

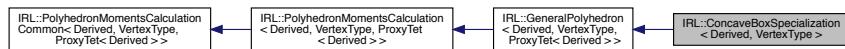
Essentially a std::vector with assertions on access through operator[] in debug mode.

The documentation for this class was generated from the following file:

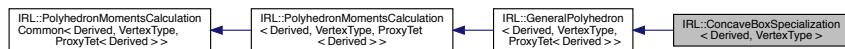
- collection.h

6.111 IRL::ConcaveBoxSpecialization< Derived, VertexType > Class Template Reference

Inheritance diagram for IRL::ConcaveBoxSpecialization< Derived, VertexType >:



Collaboration diagram for IRL::ConcaveBoxSpecialization< Derived, VertexType >:



Public Member Functions

- [HalfEdgePolyhedron< VertexType > generateHalfEdgeVersion](#) (void) const
- template<class HalfEdgePolyhedronType >
void [setHalfEdgeVersion](#) (HalfEdgePolyhedronType *a_half_edge_version) const
- [ProxyTet< Derived > getSimplexFromDecomposition](#) (const UnsignedIndex_t a_tet) const

Static Public Member Functions

- static constexpr std::array< UnsignedIndex_t, 4 > [getSimplexIndicesFromDecomposition](#) (const UnsignedIndex_t a_tet)
- static constexpr UnsignedIndex_t [getNumberOfSimplicesInDecomposition](#) (void)

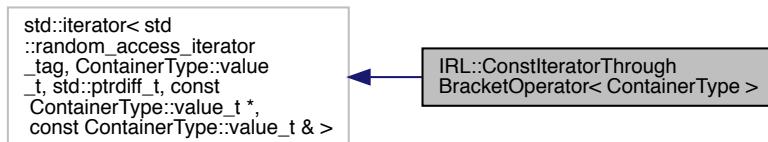
Additional Inherited Members

The documentation for this class was generated from the following file:

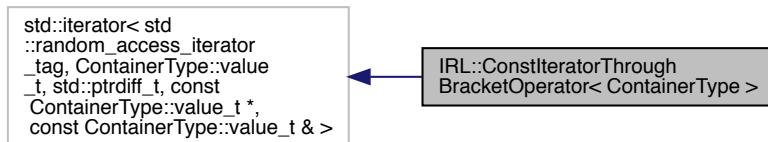
- concave_box.h

6.112 IRL::ConstIteratorThroughBracketOperator< ContainerType > Class Template Reference

Inheritance diagram for IRL::ConstIteratorThroughBracketOperator< ContainerType >:



Collaboration diagram for IRL::ConstIteratorThroughBracketOperator< ContainerType >:



Public Types

- using **value_t** = typename ContainerType::value_t

Public Member Functions

- **ConstIteratorThroughBracketOperator** (const ContainerType &a_container, const UnsignedIndex_t a_location)
- const value_t & **operator*** (void)
- const value_t & **operator->** (void)
- const value_t & **operator[]** (const std::ptrdiff_t a_index)
- **ConstIteratorThroughBracketOperator** & **operator++** (void)
- **ConstIteratorThroughBracketOperator** & **operator--** (void)
- **ConstIteratorThroughBracketOperator** **operator++** (int a_dummy_for_postfix)
- **ConstIteratorThroughBracketOperator** **operator--** (int a_dummy_for_postfix)
- **ConstIteratorThroughBracketOperator** & **operator+=** (const std::ptrdiff_t a_shift)
- **ConstIteratorThroughBracketOperator** & **operator-=** (const std::ptrdiff_t a_shift)
- std::ptrdiff_t **operator-** (const **ConstIteratorThroughBracketOperator** &a_rhs)
- bool **operator==** (const **ConstIteratorThroughBracketOperator** &a_rhs) const
- bool **operator!=** (const **ConstIteratorThroughBracketOperator** &a_rhs) const
- bool **operator<** (const **ConstIteratorThroughBracketOperator** &a_rhs) const
- bool **operator>** (const **ConstIteratorThroughBracketOperator** &a_rhs) const
- bool **operator>=** (const **ConstIteratorThroughBracketOperator** &a_rhs) const
- bool **operator<=** (const **ConstIteratorThroughBracketOperator** &a_rhs) const

Private Attributes

- const ContainerType & **container_m**
- UnsignedIndex_t **location_m**

The documentation for this class was generated from the following file:

- iterator_through_bracket_operator.h

6.113 f_tet_class::construct Interface Reference

Public Member Functions

- subroutine **tet_class_construct** (this, a_Tet_pts)

The documentation for this interface was generated from the following file:

- [f_tet_class.f90](#)

6.114 f_tri_class::construct Interface Reference

Public Member Functions

- subroutine **tri_class_construct** (this, a_pts)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.115 f_cappeddodecahedron_class::construct Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_class_construct** (this, a_dodecahedron)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.116 f_dodecahedron_class::construct Interface Reference

Public Member Functions

- subroutine **dodecahedron_class_construct** (this, a_transported_cell)

The documentation for this interface was generated from the following file:

- [f_dodecahedron_class.f90](#)

6.117 f_cappeddodecahedron_doubles3_class::construct Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_doubles3_class_construct** (this, a_dodecahedron, a_attached_data)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.118 f_polygon_class::construct Interface Reference

Public Member Functions

- subroutine **polygon_class_construct** (this, a_npts, a_pts)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.119 f_polyhedron24_class::construct Interface Reference

Public Member Functions

- subroutine **polyhedron24_class_construct** (this, a_polyhedron24)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.120 f_polyhedron24_doubles3_class::construct Interface Reference

Public Member Functions

- subroutine **polyhedron24_doubles3_class_construct** (this, a_polyhedron24, a_data)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.121 f_dividedpolygon_class::construct Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_construct** (this, a_npts, a_pts)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.122 f_rectangularcuboid_class::construct Interface Reference

Public Member Functions

- subroutine **rectangularcuboid_class_construct** (this, a_transported_cell)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.123 f_sepvm_class::construct Interface Reference

Public Member Functions

- subroutine **sepvm_class_construct** (this, a_moments_list)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.124 f_rectangularcuboid_class::construct_2pt Interface Reference

Public Member Functions

- subroutine **rectangularcuboid_class_construct_2pt** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.125 f_dividedpolygon_class::constructfrompolygon Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_constructfrompolygon** (this, a_polygon)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.126 IRL::ContinueDividingVolumeByPlanarReconstruction< HowToCutSimplexByNextPlane, HowToBranchEarlyIfNoIntersection, HowToGenerateAndHandleSimplicesFromVolumeUnderPlane, HowToGenerateAndHandleSimplicesFromVolumeAbovePlane > Class Template Reference

Static Public Member Functions

- template<class SimplexType , class ReconstructionType , class ReturnType >
static void **continueDividingVolumeByPlanarReconstruction** (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, ReturnType *a_moments_to_return)

Static Private Member Functions

- template<class SimplexType , class ReconstructionType >
static std::array< double, [SimplexWrapper< SimplexType >::simplex_nvert](#) > **cutSimplexByNextPlane** (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, LookupIndex_t *a_cutting_case_for_simplex_and_current_plane)
- template<class SimplexType , class ReconstructionType , class ReturnType >
static bool **earlyBranchingIfSimplexIsNotIntersectedByCurrentPlane** (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, const LookupIndex_t a_cutting_case, ReturnType *a_moments_to_return)
- template<class SimplexType , class ReconstructionType , class VerticesList , class ReturnType >
static void **generateAndHandleSimplicesFromVolumeUnderPlane** (const ReconstructionType &a_reconstruction, const LookupIndex_t a_cutting_case_for_simplex_and_current_plane, const VerticesList &a_simplex_vertices_and_intersection_points, const UnsignedIndex_t a_cutting_plane_index, ReturnType *a_moments_to_return)
- template<class SimplexType , class ReconstructionType , class VerticesList , class ReturnType >
static void **generateAndHandleSimplicesFromVolumeAbovePlane** (const ReconstructionType &a_reconstruction, const LookupIndex_t a_cutting_case_for_simplex_and_current_plane, const VerticesList &a_simplex_vertices_and_intersection_points, const UnsignedIndex_t a_index_for_plane_that_just_cut, ReturnType *a_moments_to_return)

The documentation for this class was generated from the following file:

- [continue_dividing_volume.h](#)

6.127 IRL::ContinueReducingVolumeToBeInternalToReconstruction Struct Reference

Static Public Member Functions

- template<class SimplexType , class ReconstructionType , class VerticesList , class ReturnType >
static void generateAndHandleSimplicesFromVolumeUnderPlane (const ReconstructionType &a_reconstruction, const LookupIndex_t a_cutting_case_for_simplex_and_current_plane, const VerticesList &a_simplex_vertices_and_intersection_points, const UnsignedIndex_t a_cutting_plane_index, ReturnType *a_moments_to_return)

The documentation for this struct was generated from the following file:

- continue_dividing_volume.h

6.128 f_planarseparator_class::copy Interface Reference

Public Member Functions

- subroutine **planarseparator_class_copy** (this, a_other_PlanarSeparator)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.129 IRL::CutSimplexByNextPlaneAccessedByIndex Struct Reference

Static Public Member Functions

- template<class SimplexType , class ReconstructionType >
static std::array< double, SimplexWrapper< SimplexType >::simplex_nvert > cutSimplexByNextPlane (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, LookupIndex_t *a_cutting_case_for_simplex_and_current_plane)

The documentation for this struct was generated from the following file:

- continue_dividing_volume.h

6.130 IRL::CutTetVertices< VertexType > Class Template Reference

Public Types

- using **pt_type** = VertexType

Public Member Functions

- **CutTetVertices** (const pt_type &a_pt_0, const pt_type &a_pt_1, const pt_type &a_pt_2, const pt_type &a_pt_3)
- pt_type & **operator[]** (const UnsignedIndex_t a_index)
- const pt_type & **operator[]** (const UnsignedIndex_t a_index) const

Private Attributes

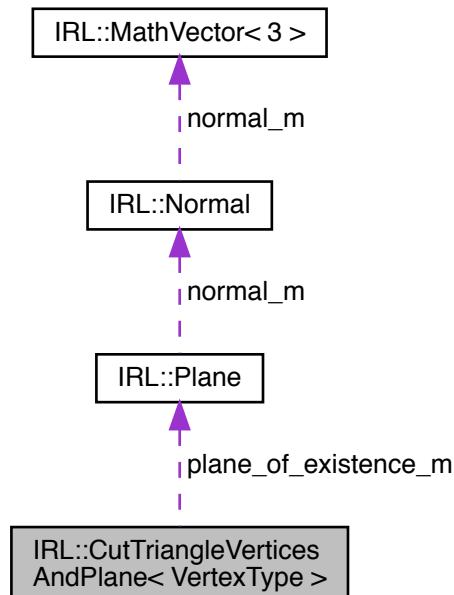
- std::array< pt_type, 8 > **vertices_m**

The documentation for this class was generated from the following file:

- simplex_wrapper.h

6.131 IRL::CutTriangleVerticesAndPlane< VertexType > Class Template Reference

Collaboration diagram for IRL::CutTriangleVerticesAndPlane< VertexType >:



Public Types

- using **pt_type** = VertexType

Public Member Functions

- **CutTriangleVerticesAndPlane** (const pt_type &a_pt_0, const pt_type &a_pt_1, const pt_type &a_pt_2, const Plane *a_plane_of_existence)
- pt_type & **operator[]** (const UnsignedIndex_t a_index)
- const pt_type & **operator[]** (const UnsignedIndex_t a_index) const
- const Plane & **getPlaneOfExistence** (void) const

Private Attributes

- std::array< pt_type, 5 > **vertices_m**
- const Plane * **plane_of_existence_m**

The documentation for this class was generated from the following file:

- simplex_wrapper.h

6.132 f_bytebuffer_class::dataptr Interface Reference

Public Member Functions

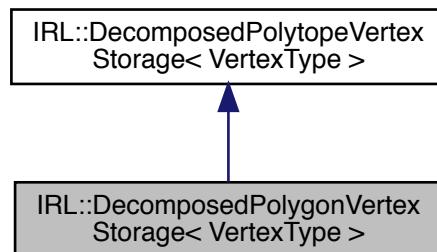
- integer(irl_byte_t) function, dimension(:,), pointer **bytebuffer_class_dataptr** (this)

The documentation for this interface was generated from the following file:

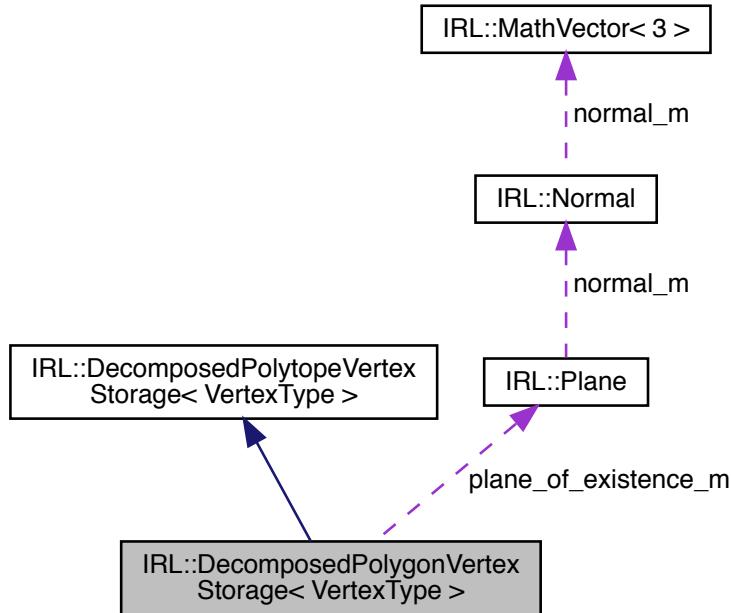
- [f_bytebuffer_class.f90](#)

6.133 IRL::DecomposedPolygonVertexStorage< VertexType > Class Template Reference

Inheritance diagram for IRL::DecomposedPolygonVertexStorage< VertexType >:



Collaboration diagram for IRL::DecomposedPolygonVertexStorage< VertexType >:



Public Member Functions

- template<class GeometryType >
void **resetFromGeometry** (const GeometryType &a_geometry)
- template<class BasePtType >
void **resetFromGeometry** (const ExpandableDividedPolygon< BasePtType > &a_geometry)
- void **setPlaneOfExistence** (const [Plane](#) &a_plane)
- const [Plane](#) & **getPlaneOfExistence** (void) const

Private Attributes

- [Plane](#) **plane_of_existence_m**

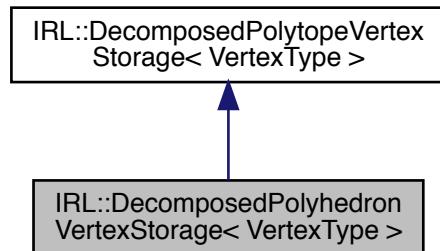
Additional Inherited Members

The documentation for this class was generated from the following file:

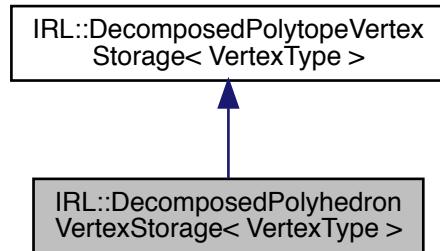
- decomposed_polytope_vertex_storage.h

6.134 IRL::DecomposedPolyhedronVertexStorage< VertexType > Class Template Reference

Inheritance diagram for IRL::DecomposedPolyhedronVertexStorage< VertexType >:



Collaboration diagram for IRL::DecomposedPolyhedronVertexStorage< VertexType >:



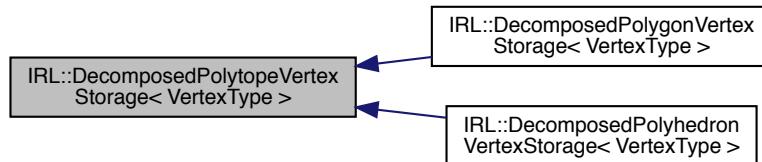
Additional Inherited Members

The documentation for this class was generated from the following file:

- `decomposed_polytope_vertex_storage.h`

6.135 IRL::DecomposedPolytopeVertexStorage< VertexType > Class Template Reference

Inheritance diagram for IRL::DecomposedPolytopeVertexStorage< VertexType >:



Public Types

- using **pt_type** = VertexType

Public Member Functions

- template<class GeometryType >
void **resetFromGeometry** (const GeometryType &a_geometry)
- pt_type & **operator[]** (const UnsignedIndex_t a_index)
- const pt_type & **operator[]** (const UnsignedIndex_t a_index) const
- void **push_back** (const VertexType a_vertex)
- UnsignedIndex_t **size** (void) const
- void **resize** (const UnsignedIndex_t a_new_size)
- double **getDistance** (const UnsignedIndex_t a_index) const
- double **setDistance** (const UnsignedIndex_t a_index, const double a_distance)

Private Attributes

- std::vector< VertexType > **vertex_storage_m**
- std::vector< double > **distance_m**

The documentation for this class was generated from the following file:

- decomposed_polytope_vertex_storage.h

6.136 IRL::DelayedExpandableVertexAccess< Derived, VertexType, kMaskedVertices >

Class Template Reference

Public Member Functions

- **DelayedExpandableVertexAccess** (std::initializer_list< VertexType > a_list)
- VertexType & **access** (const UnsignedIndex_t a_index)
- const VertexType & **access** (const UnsignedIndex_t a_index) const
- VertexType & **unmaskedAccess** (const UnsignedIndex_t a_index)
- const VertexType & **unmaskedAccess** (const UnsignedIndex_t a_index) const
- UnsignedIndex_t **getNumberOfVerticesInObject** (void) const

Return the number of vertices in this polygon.
- void **setNumberOfVerticesInObject** (const UnsignedIndex_t a_size)
- void **addVertex** (const VertexType &a_vertex)
- void **addVertexAtIndex** (const UnsignedIndex_t a_index, const VertexType &a_pt)
- void **reserve** (const UnsignedIndex_t a_size)
- void **removeLastVertex** (void)

Add point to the point list.
- void **removePt** (const UnsignedIndex_t a_index)
- LargeOffsetIndex_t **getSerializedSize** (void) const

Return size of the serialized Polyhedron.
- void **serialize** (ByteBuffer *a_buffer) const

Serialize and pack the Polyhedron.
- void **unpackSerialized** (ByteBuffer *a_buffer)

Unpack the polyhedron.

Static Public Member Functions

- static Derived **fromRawPtPointer** (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts)

Construct n-pts form array of points.
- static Derived **fromRawDoublePointer** (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs)

Construct n-pts form array of doubles.

Protected Member Functions

- **DelayedExpandableVertexAccess** (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts)

Construct n-pts form array of pts.
- **DelayedExpandableVertexAccess** (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs)

Construct n-pts form array of doubles.
- void **insertMaskedVertices** (void)

Private Attributes

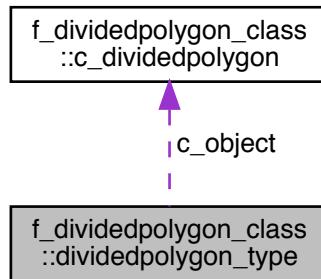
- **ExpandablePtList< VertexType > pt_list_m**

The documentation for this class was generated from the following file:

- delayed_expandable_vertex_access.h

6.137 f_dividedpolygon_class::dividedpolygon_type Type Reference

Collaboration diagram for f_dividedpolygon_class::dividedpolygon_type:



Public Member Functions

- final [dividedpolygon_class_delete](#)

Private Attributes

- type([c_dividedpolygon](#)), private **c_object**

The documentation for this type was generated from the following file:

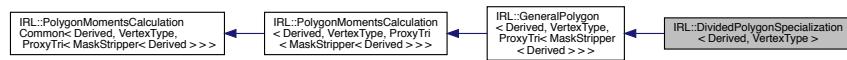
- [f_dividedpolygon_class.f90](#)

6.138 IRL::DividedPolygonSpecialization< Derived, VertexType > Class Template Reference

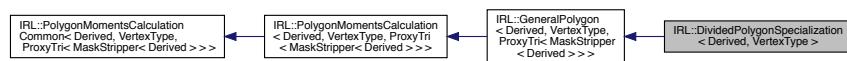
A polygon that knows its centroid and can return the triangles that it is comprised of.

```
#include <divided_polygon.h>
```

Inheritance diagram for IRL::DividedPolygonSpecialization< Derived, VertexType >:



Collaboration diagram for IRL::DividedPolygonSpecialization< Derived, VertexType >:



Public Member Functions

- UnsignedIndex_t **getNumberOfSimplicesInDecomposition** (void) const
- std::array< UnsignedIndex_t, 3 > **getSimplexIndicesFromDecomposition** (UnsignedIndex_t a_tri_number_to_get) const
- **ProxyTri< MaskStripper< Derived > > getSimplexFromDecomposition** (UnsignedIndex_t a_tri_number_to_get) const
Returns the triangle.
- **HalfEdgePolygon< VertexType > generateHalfEdgeVersion** (void) const
- template<class HalfEdgePolygonClass>
void setHalfEdgeVersion (HalfEdgePolygonClass *a_half_edge_version) const
- void **resetCentroid** (void)
- **Pt calculateCentroid** (void) const
- **VolumeMoments calculateMoments** (void) const

Protected Member Functions

- VertexType & **getCentroid** (void)
- const VertexType & **getCentroid** (void) const

Private Member Functions

- Derived & **getDerived** (void)
- const Derived & **getDerived** (void) const
- **Pt calculatePolygonCentroid** (void) const

Additional Inherited Members

6.138.1 Detailed Description

```
template<class Derived, class VertexType>
class IRL::DividedPolygonSpecialization< Derived, VertexType >
```

A polygon that knows its centroid and can return the triangles that it is comprised of.

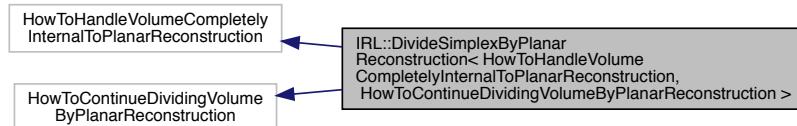
The documentation for this class was generated from the following file:

- divided_polygon.h

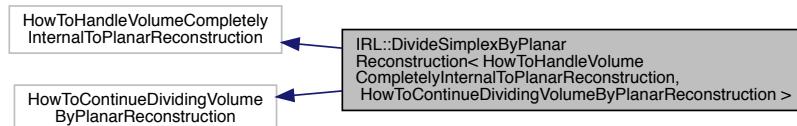
6.139 IRL::DivideSimplexByPlanarReconstruction<
 HowToHandleVolumeCompletelyInternalToPlanarReconstruction,
 HowToContinueDividingVolumeByPlanarReconstruction > Class Template Reference 191
6.139 IRL::DivideSimplexByPlanarReconstruction ← HowToHandleVolumeCompletely →

InternalToPlanarReconstruction, HowToContinueDividingVolumeByPlanarReconstruction
 > Class Template Reference

Inheritance diagram for IRL::DivideSimplexByPlanarReconstruction< HowToHandleVolumeCompletelyInternalToPlanarReconstruction, HowToContinueDividingVolumeByPlanarReconstruction >:



Collaboration diagram for IRL::DivideSimplexByPlanarReconstruction< HowToHandleVolumeCompletelyInternalToPlanarReconstruction, HowToContinueDividingVolumeByPlanarReconstruction >:



Static Public Member Functions

- template<class SimplexType , class ReconstructionType , class ReturnType >
 static void **execute** (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, ReturnType *a_moments_to_return)

Static Private Member Functions

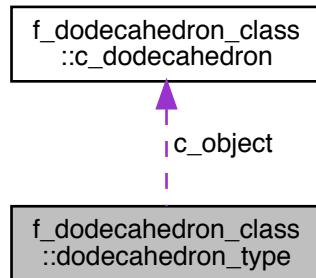
- template<class ReconstructionType >
 static bool **noPlanesLeftToDivideBy** (const UnsignedIndex_t a_cutting_plane_index, const ReconstructionType &a_reconstruction)
- template<class SimplexType , class ReconstructionType , class ReturnType >
 static void **handleVolumeCompletelyInternalToPlanarReconstruction** (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, ReturnType *a_moments_to_return)
- template<class SimplexType , class ReconstructionType , class ReturnType >
 static void **continueDividingVolumeByPlanarReconstruction** (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, ReturnType *a_moments_to_return)

The documentation for this class was generated from the following file:

- cut_simplex_drivers.h

6.140 f_dodecahedron_class::dodecahedron_type Type Reference

Collaboration diagram for f_dodecahedron_class::dodecahedron_type:



Public Member Functions

- final **dodecahedron_class_delete**

Private Attributes

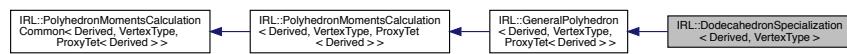
- type([c_dodecahedron](#)), private **c_object**

The documentation for this type was generated from the following file:

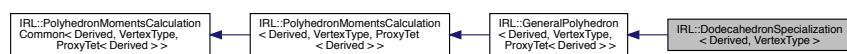
- [f_dodecahedron_class.f90](#)

6.141 IRL::DodecahedronSpecialization< Derived, VertexType > Class Template Reference

Inheritance diagram for IRL::DodecahedronSpecialization< Derived, VertexType >:



Collaboration diagram for IRL::DodecahedronSpecialization< Derived, VertexType >:



Public Member Functions

- `HalfEdgePolyhedron< VertexType > generateHalfEdgeVersion (void) const`
- `template<class HalfEdgePolyhedronType>`
`void setHalfEdgeVersion (HalfEdgePolyhedronType *a_half_edge_version) const`
- `ProxyTet< Derived > getSimplexFromDecomposition (const UnsignedIndex_t a_tet) const`

Static Public Member Functions

- `static constexpr UnsignedIndex_t getNumberOfSimplicesInDecomposition (void)`
- `static constexpr std::array< UnsignedIndex_t, 4 > getSimplexIndicesFromDecomposition (const UnsignedIndex_t a_tet)`

Additional Inherited Members

The documentation for this class was generated from the following file:

- `dodecahedron.h`

6.142 IRL::DoesNotHaveACollection< ReturnType > Struct Template Reference

Static Public Attributes

- `static constexpr bool value = !HasACollection<ReturnType>::value`

The documentation for this struct was generated from the following file:

- `class_classifications.h`

6.143 IRL::DoesNotHaveALocalizer< ReconstructionType > Struct Template Reference

Static Public Attributes

- `static constexpr bool value = !HasALocalizer<ReconstructionType>::value`

The documentation for this struct was generated from the following file:

- `class_classifications.h`

6.144 IRL::DoesNotHaveANestedType< ReturnType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value** = !HasANestedType<ReturnType>::value

The documentation for this struct was generated from the following file:

- class_classifications.h

6.145 IRL::DoesNotHaveAReconstructionLink< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value**

6.145.1 Member Data Documentation

6.145.1.1 value

```
template<class ReconstructionType >
constexpr bool IRL::DoesNotHaveAReconstructionLink< ReconstructionType >::value [static]
```

Initial value:

```
=  
    !HasAReconstructionLink<ReconstructionType>::value
```

The documentation for this struct was generated from the following file:

- class_classifications.h

6.146 IRL::DoesNotHaveASeparator< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value** = !HasASeparator<ReconstructionType>::value

The documentation for this struct was generated from the following file:

- class_classifications.h

6.147 IRL::doNoEarlyBranching Struct Reference

Static Public Member Functions

- template<class SimplexType , class ReconstructionType , class ReturnType >
static constexpr bool **earlyBranchingIfSimplexIsNotIntersectedByCurrentPlane** (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, const LookupIndex_t a_cutting_case, ReturnType *a_moments_to_return)

The documentation for this struct was generated from the following file:

- continue_dividing_volume.h

6.148 IRL::earlyBranchBothFullyBelowAndFullyAboveSimplicesLink Struct Reference

Static Public Member Functions

- template<class SimplexType , class ReconstructionType , class ReturnType >
static bool **earlyBranchingIfSimplexIsNotIntersectedByCurrentPlane** (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, const LookupIndex_t a_cutting_case, ReturnType *a_moments_to_return)

The documentation for this struct was generated from the following file:

- continue_dividing_volume.h

6.149 IRL::earlyBranchBothFullyBelowAndFullyAboveSimplicesNoLink Struct Reference

Static Public Member Functions

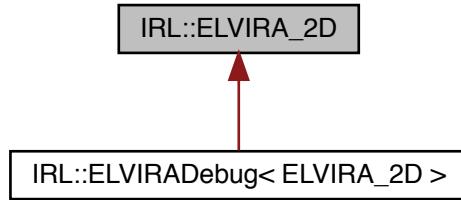
- template<class SimplexType , class ReconstructionType , class ReturnType >
static bool **earlyBranchingIfSimplexIsNotIntersectedByCurrentPlane** (const SimplexType &a_simplex, const ReconstructionType &a_reconstruction, const UnsignedIndex_t a_cutting_plane_index, const LookupIndex_t a_cutting_case, ReturnType *a_moments_to_return)

The documentation for this struct was generated from the following file:

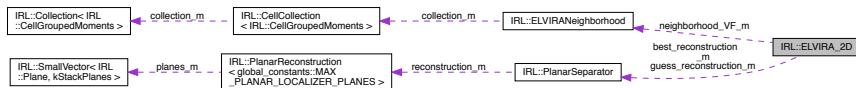
- continue_dividing_volume.h

6.150 IRL::ELVIRA_2D Class Reference

Inheritance diagram for IRL::ELVIRA_2D:



Collaboration diagram for IRL::ELVIRA_2D:



Public Member Functions

- [ELVIRA_2D \(void\)=default](#)
Default constructor.
- [PlanarSeparator solve \(const ELVIRANeighborhood *a_neighborhood_pointer\)](#)
Solve the system for the reconstruction, restarting the neighboring geometry.
- [~ELVIRA_2D \(void\)=default](#)
Default destructor.

Private Member Functions

- [PlanarSeparator solve \(void\)](#)
Solve the system for the reconstruction.
- [double & columnSum \(const int i\)](#)
Return value of column sum at index. range for index of -1:1, with center cell column being at index 0,0.
- [Pt & columnCenters \(const int i\)](#)
Return point of column center at index i,k, with the integrated dimensions index being 0. range for index of -1:1, with center cell column being at index 0,0.
- [double columnCenters \(const UnsignedIndex_t dimension, const int i\)](#)
Overloaded column centers to return the location of the column center for the dimension.
- [UnsignedIndex_t calculateLinearIndex \(const int i\)](#)
- [void fillOtherNormals \(const UnsignedIndex_t a_dimension_index, double a_normal_approximation_< holder\[3\]>\)](#)
- [void makeColumnSumsX \(void\)](#)

- `void makeColumnSumsX (void)`

Make column sums from summing in the x direction.
- `double computeDerivative0 (const int a_left_index, const int a_right_index, const UnsignedIndex_t spatial_index)`

Compute derivative of the column heights.
- `void tryNormal (Normal a_normal)`

Calculate error given the supplied normal. If lowest error yet, save as best_reconstruction_m.

Private Attributes

- `friend ELVIRADebug< ELVIRA_2D >`
- `const ELVIRANeighborhood * neighborhood_VF_m`

Storage of the stencil information.
- `std::array< double, 3 > column_sums_m`

Array of column sums needed in ELVIRA.
- `std::array< Pt, 3 > column_centers_m`

Array of points for centers of columns. Needed for derivatives.
- `PlanarSeparator guess_reconstruction_m`

Best reconstruction found so far.
- `PlanarSeparator best_reconstruction_m`

Best reconstruction found so far.
- `double minimum_error_m`

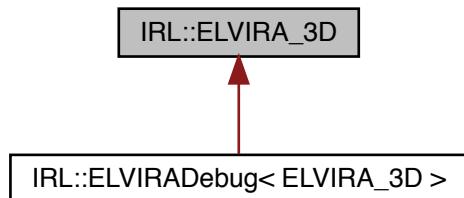
Lowest error resulting from best reconstruction.

The documentation for this class was generated from the following files:

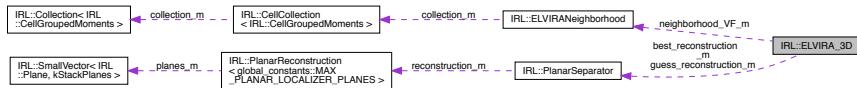
- `elvira.h`
- `elvira.cpp`

6.151 IRL::ELVIRA_3D Class Reference

Inheritance diagram for IRL::ELVIRA_3D:



Collaboration diagram for IRL::ELVIRA_3D:



Public Member Functions

- **ELVIRA_3D (void)=default**
Default constructor.
- **PlanarSeparator solve (const ELVIRANeighborhood *a_neighborhood_pointer)**
Solve the system for the reconstruction, restarting the neighboring geoemtry.
- **~ELVIRA_3D (void)=default**
Default destructor.

Private Member Functions

- **PlanarSeparator solve (void)**
Solve the system for the reconstruction.
- **double & columnSum (const int i, const int j)**
Return value of column sum at index. range for index of -1:1, with center cell column being at index 0,0.
- **Pt & columnCenters (const int i, const int j)**
Return point of column center at index i,k, with the integrated dimensions index being 0. range for index of -1:1, with center cell column being at index 0,0.
- **double columnCenters (const UnsignedIndex_t dimension, const int i, const int j)**
Overloaded column centers to return the location of the column center for the dimension.
- **UnsignedIndex_t calculateLinearIndex (const int i, const int j)**
- **void fillOtherNormals (const UnsignedIndex_t a_first_column_index, const UnsignedIndex_t a_second_column_index, double a_normal_approximation_holder[3][2])**
Fill in normal approximations from derivatives of column heights.
- **void makeColumnSumsX (void)**
Make column sums from summing in the x direction.
- **void makeColumnSumsY (void)**
Make column sums from summing in the y direction.
- **void makeColumnSumsZ (void)**
Make column sums from summing in the z direction.
- **double computeDerivative0 (const int a_left_index, const int a_right_index, const UnsignedIndex_t a_spatial_index)**
Compute derivative of first dimension of column heights.
- **double computeDerivative1 (const int a_left_index, const int a_right_index, const UnsignedIndex_t a_spatial_index)**
Compute derivative of second dimension of column heights.
- **void tryNormal (Normal a_normal)**
Calculate error given the supplied normal. If lowest error yet, save as best_reconstruction_m.

Private Attributes

- friend **ELVIRADebug< ELVIRA_3D >**
- const **ELVIRANeighborhood * neighborhood_VF_m**
Storage of the stencil information.
- std::array< double, 9 > **column_sums_m**
Array of column sums needed in ELVIRA.
- std::array< Pt, 9 > **column_centers_m**
Array of points for centers of columns. Needed for derivatives.
- **PlanarSeparator guess_reconstruction_m**
Best reconstruction found so far.
- **PlanarSeparator best_reconstruction_m**
Best reconstruction found so far.
- double **minimum_error_m**
Lowest error resulting from best reconstruction.

The documentation for this class was generated from the following files:

- elvira.h
- elvira.cpp

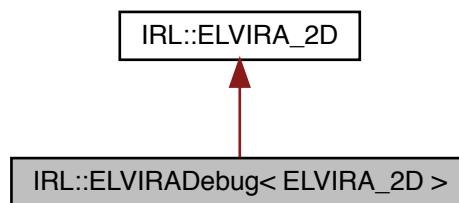
6.152 IRL::ELVIRADebug< BaseELVIRA > Class Template Reference

The documentation for this class was generated from the following file:

- elvira.h

6.153 IRL::ELVIRADebug< ELVIRA_2D > Class Template Reference

Inheritance diagram for IRL::ELVIRADebug< ELVIRA_2D >:



Collaboration diagram for IRL::ELVIRADebug< ELVIRA_2D >:



Public Member Functions

- **ELVIRADebug** (void)=default
Default constructor.
- **PlanarSeparator solve** (const **ELVIRANeighborhood** *a_neighborhood_pointer)
Solve the system for the reconstruction, restarting the neighboring geoemtry.

Private Member Functions

- **PlanarSeparator solve** (void)
Solve the system for the reconstruction.
- void **writeOutVolumeFractions** (void)
Write out the centroids and weights to enable visualization of what optimization is driving towards.
- void **writeOutPlane** (**Normal** a_normal, const std::string &a_prefix, const UnsignedIndex_t a_iteration_number)
- void **tryNormal** (**Normal** a_normal)

Private Attributes

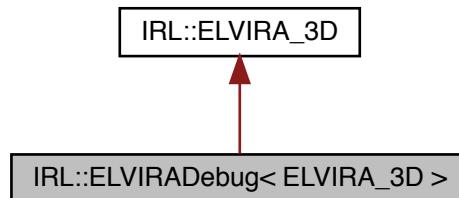
- UnsignedIndex_t **counter_m** = 0
Counter to note which attempted normal it is.

The documentation for this class was generated from the following files:

- elvira.h
- elvira.cpp

6.154 IRL::ELVIRADebug< ELVIRA_3D > Class Template Reference

Inheritance diagram for IRL::ELVIRADebug< ELVIRA_3D >:



Collaboration diagram for IRL::ELVIRADebug< ELVIRA_3D >:



Public Member Functions

- `ELVIRADebug (void)=default`
Default constructor.
- `PlanarSeparator solve (const ELVIRANeighborhood *a_neighborhood_pointer)`
Solve the system for the reconstruction, restarting the neighboring geoemtry.

Private Member Functions

- `PlanarSeparator solve (void)`
Solve the system for the reconstruction.
- `void writeOutVolumeFractions (void)`
Write out the centroids and weights to enable visualization of what optimization is driving towards.
- `void writeOutPlane (Normal a_normal, const std::string &a_prefix, const UnsignedIndex_t a_iteration_← number)`
- `void tryNormal (Normal a_normal)`

Private Attributes

- `UnsignedIndex_t counter_m = 0`
Counter to note which attempted normal it is.

The documentation for this class was generated from the following files:

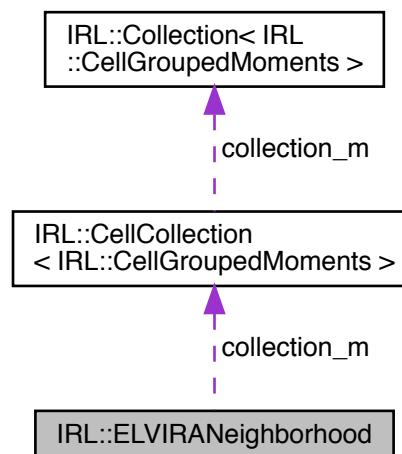
- elvira.h
- elvira.cpp

6.155 IRL::ELVIRANeighborhood Class Reference

Below plane volume fraction information in a stencil, to be used for ELVIRA. Up to 27 cells to cover 3x3x3 stencil in 3D. Can partially fill to 9 for 2D.

```
#include <elvira_neighborhood.h>
```

Collaboration diagram for IRL::ELVIRANeighborhood:



Public Member Functions

- `ELVIRANeighborhood (void)=default`
Default constructor.
- `void setMember (const RectangularCuboid *a_rectangular_cuboid, const double *a_liquid_volume_fraction, const int i, const int j, const int k=-1)`
Construct a `CellGroupedMoments` and add it to the collection for index i,j,k.
- `const RectangularCuboid & getCell (const int i, const int j, const int k=-1) const`
Return the cell stored at the index i,j,k.
- `double getStoredMoments (const int i, const int j, const int k=-1) const`
Return moments stored at the index i,j,k.
- `void resize (const UnsignedIndex_t a_size)`
Set size of the neighborhood.
- `~ELVIRANeighborhood (void)=default`
Default destructor.

Private Types

- `using CGD = CellGroupedMoments< RectangularCuboid, double >`

Private Member Functions

- `UnsignedIndex_t calculateLinearIndex (const int i, const int j, const int k) const`
Calculate linear index from i,j,k.

Private Attributes

- `CellCollection< CGD > collection_m`
Collection that holds correct moments.

6.155.1 Detailed Description

Below plane volume fraction information in a stencil, to be used for ELVIRA. Up to 27 cells to cover 3x3x3 stencil in 3D. Can partially fill to 9 for 2D.

This class is used to store neighboring volume fraction information and RectangularCuboid cells to be used with the ELVIRA. There is an assumed order of the cells that will be added, with x being the fastest dimension, y being second, and z being slowest. i.e. for the loop order:

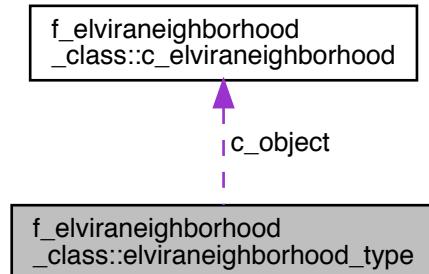
`loop z loop y loop x (*this)[] = (...)`

The documentation for this class was generated from the following files:

- `elvira_neighborhood.h`
- `elvira_neighborhood.cpp`

6.156 f_elviraneighborhood_class::elviraneighborhood_type Type Reference

Collaboration diagram for f_elviraneighborhood_class::elviraneighborhood_type:



Public Member Functions

- final **elviraneighborhood_class_delete**

Private Attributes

- type([c_elviraneighborhood](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_elviraneighborhood_class.f90](#)

6.157 f_r2pneighborhood_rectangularcuboid_class::emptyneighborhood Interface Reference

Public Member Functions

- subroutine **r2pneighborhood_rectangularcuboid_class_emptyneighborhood** (this)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.158 f_lviraneighborhood_rectangularcuboid_class::emptyneighborhood Interface Reference

Public Member Functions

- subroutine **lviraneighborhood_rectangularcuboid_class_emptyneighborhood** (this)

The documentation for this interface was generated from the following file:

- `f_lviraneighborhood_rectangularcuboid_class.f90`

6.159 f_listedvm_vman_class::erase Interface Reference

Public Member Functions

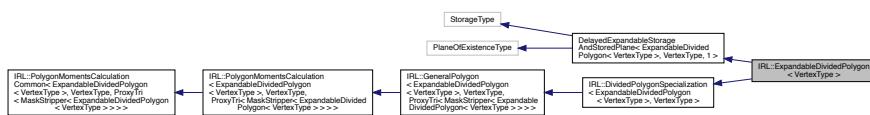
- subroutine **listedvm_vman_class_erase** (this, a_index)

The documentation for this interface was generated from the following file:

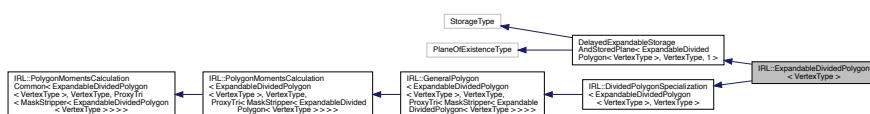
- `f_listedvm_vman_class.f90`

6.160 IRL::ExpandableDividedPolygon< VertexType > Class Template Reference

Inheritance diagram for IRL::ExpandableDividedPolygon< VertexType >:



Collaboration diagram for IRL::ExpandableDividedPolygon< VertexType >:



Static Public Member Functions

- template<class AnyDerivedClass , class AnySimplexType >
 static **ExpandableDividedPolygon** **fromPolygon** (const **GeneralPolygon< AnyDerivedClass, VertexType, AnySimplexType >** &a_polygon)

Private Member Functions

- template<class AnyDerivedClass , class AnySimplexType >
ExpandableDividedPolygon (const [GeneralPolygon](#)< AnyDerivedClass, VertexType, AnySimplexType > &a_polygon)

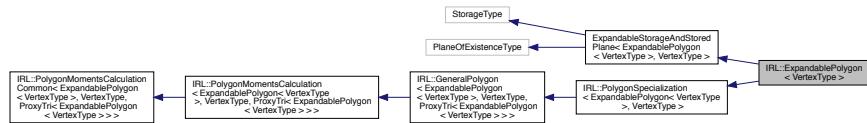
Additional Inherited Members

The documentation for this class was generated from the following file:

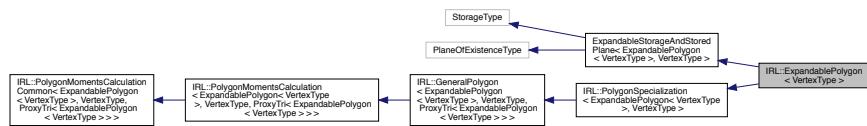
- divided_polygon.h

6.161 IRL::ExpandablePolygon< VertexType > Class Template Reference

Inheritance diagram for IRL::ExpandablePolygon< VertexType >:



Collaboration diagram for IRL::ExpandablePolygon< VertexType >:



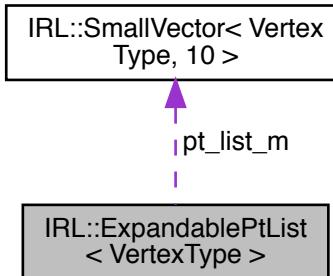
Additional Inherited Members

The documentation for this class was generated from the following file:

- polygon.h

6.162 IRL::ExpandablePtList< VertexType > Class Template Reference

Collaboration diagram for IRL::ExpandablePtList< VertexType >:



Public Types

- using **StorageType** = `SmallVector< VertexType, 10 >`
- using **iterator** = typename StorageType::iterator
- using **const_iterator** = typename StorageType::const_iterator

Public Member Functions

- `ExpandablePtList (void)=default`
Default constructor.
- `UnsignedIndex_t getNumberOfPts (void) const`
Const return the number of vertices in this polygon.
- `void setNumberOfPts (const UnsignedIndex_t a_number)`
Set the number of vertices directly.
- `void reserve (const UnsignedIndex_t a_number)`
Reserve space in the Pt vector.
- `void addPt (const VertexType &a_pt)`
Add point to the point list.
- `void addPtAtIndex (const UnsignedIndex_t a_index, const VertexType &a_pt)`
Add point to the point list.
- `VertexType & operator[] (const UnsignedIndex_t a_index)`
Access through overloaded operator[].
- `const VertexType & operator[] (const UnsignedIndex_t a_index) const`
Const access through overloaded operator[].
- `void removePt (const UnsignedIndex_t a_index)`
Remove point from the point list.
- `void removeLastPt (void)`
Remove last point from the point list.
- `IRL::Pt getLowerLimits (void) const`
Return a point for the lower limits of the polygon in 3D space.

- `IRL::Pt getUpperLimits (void) const`
Return a point for the upper limits of the polygon in 3D space.
- iterator `begin (void) noexcept`
- const_iterator `begin (void) const noexcept`
- const_iterator `cbegin (void) const noexcept`
- iterator `end (void) noexcept`
- const_iterator `end (void) const noexcept`
- const_iterator `cend (void) const noexcept`
- LargeOffsetIndex_t `getSerializedSize (void) const`
Return size of the serialized `ExpandablePtList`.
- void `serialize (ByteBuffer *a_buffer) const`
Serialize and pack the points.
- void `unpackSerialized (ByteBuffer *a_buffer)`
Unpack the points and store.
- `~ExpandablePtList (void)=default`
Default destructor.

Static Public Member Functions

- static `ExpandablePtList fromRawPtPointer (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts)`
- static `ExpandablePtList fromRawDoublePointer (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs)`

Protected Member Functions

- `ExpandablePtList (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts)`
Construct n-pts form array of points.
- `ExpandablePtList (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs)`
Construct n-pts form array of doubles.

Private Member Functions

- void `checkIfStaticAllocationExceeded (void) const`

Private Attributes

- `StorageType pt_list_m`

Friends

- class `PolygonBase< VertexType >`

The documentation for this class was generated from the following file:

- `expandable_pt_list.h`

6.163 IRL::Expr< E > Struct Template Reference

Public Member Functions

- **operator const E & () const**

The documentation for this struct was generated from the following file:

- [helper.h](#)

6.164 f_bytebuffer_class::F_ByteBuffer_dataPtr Interface Reference

Public Member Functions

- type(c_ptr) function **f_bytebuffer_dataptr (this)**

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.165 f_bytebuffer_class::F_ByteBuffer_delete Interface Reference

Public Member Functions

- subroutine **f_bytebuffer_delete (this)**

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.166 f_bytebuffer_class::F_ByteBuffer_getSize Interface Reference

Public Member Functions

- integer(c_size_t) function **f_bytebuffer_getsize (this)**

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.167 `f_bytebuffer_class::F_ByteBuffer_new` Interface Reference

Public Member Functions

- subroutine `f_bytebuffer_new` (this)

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.168 `f_bytebuffer_class::F_ByteBuffer_resetBufferPointer` Interface Reference

Public Member Functions

- subroutine `f_bytebuffer_resetbufferpointer` (this)

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.169 `f_bytebuffer_class::F_ByteBuffer_setSize` Interface Reference

Public Member Functions

- subroutine `f_bytebuffer_setsize` (this, a_size)

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.170 `f_cappeddodecahedron_class::F_CappedDodecahedron_adjustCapToMatchVolume` Interface Reference

Public Member Functions

- subroutine `f_cappeddodecahedron_adjustcaptomatchvolume` (this, a_correct_signed_volume)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.171 f_cappeddodecahedron_class::F_CappedDodecahedron_construct Interface Reference

Public Member Functions

- subroutine [f_cappeddodecahedron_construct](#) (this, a_dodecahedron)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.172 f_cappeddodecahedron_class::F_CappedDodecahedron_delete Interface Reference

Public Member Functions

- subroutine [f_cappeddodecahedron_delete](#) (this)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.173 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_← adjustCapToMatchVolume Interface Reference

Public Member Functions

- subroutine [f_cappeddodecahedron_doubles3_adjustcaptomatchvolume](#) (this, a_correct_signed_←
volume)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.174 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_← construct Interface Reference

Public Member Functions

- subroutine [f_cappeddodecahedron_doubles3_construct](#) (this, a_dodecahedron, a_attached_data)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

**6.175 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_←
delete Interface Reference**

Public Member Functions

- subroutine **f_cappeddodecahedron_doubles3_delete** (this)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

**6.176 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_←
getBoundingPts Interface Reference**

Public Member Functions

- subroutine **f_cappeddodecahedron_doubles3_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

**6.177 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_←
getData Interface Reference**

Public Member Functions

- subroutine **f_cappeddodecahedron_doubles3_getdata** (this, a_index, a_data)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

**6.178 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_←
getPt Interface Reference**

Public Member Functions

- subroutine **f_cappeddodecahedron_doubles3_getpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.179 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_← new Interface Reference

Public Member Functions

- subroutine **f_cappeddodecahedron_doubles3_new** (this)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.180 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_← setData Interface Reference

Public Member Functions

- subroutine **f_cappeddodecahedron_doubles3_setdata** (this, a_index, a_data)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.181 f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_← setPt Interface Reference

Public Member Functions

- subroutine **f_cappeddodecahedron_doubles3_setpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.182 f_cappeddodecahedron_class::F_CappedDodecahedron_getBoundingPts Interface Reference

Public Member Functions

- subroutine **f_cappeddodecahedron_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.183 f_cappeddodecahedron_class::F_CappedDodecahedron_getPt Interface Reference

Public Member Functions

- subroutine **f_cappeddodecahedron_getpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.184 f_cappeddodecahedron_class::F_CappedDodecahedron_new Interface Reference

Public Member Functions

- subroutine **f_cappeddodecahedron_new** (this)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.185 f_constants::F_Constants_setMinimumSurfaceAreaToTrack Interface Reference

Public Member Functions

- subroutine **f_constants_setminimumsurfaceareatotrack** (a_minimum_surface_area_to_track)

The documentation for this interface was generated from the following file:

- [f_constants.f90](#)

6.186 f_constants::F_Constants_setMinimumVolumeToTrack Interface Reference

Public Member Functions

- subroutine **f_constants_setminimumvolumetotrack** (a_minimum_volume_to_track)

The documentation for this interface was generated from the following file:

- [f_constants.f90](#)

6.187 f_constants::F_Constants_setVolumeFractionBounds Interface Reference

Public Member Functions

- subroutine **f_constants_setvolumefractionbounds** (a_VF_low)

The documentation for this interface was generated from the following file:

- [f_constants.f90](#)

6.188 f_constants::F_Constants_setVolumeFractionToleranceForDistanceFinding Interface Reference

Public Member Functions

- subroutine **f_constants_setvolumefractiontolerancefordistancefinding** (a_tolerance)

The documentation for this interface was generated from the following file:

- [f_constants.f90](#)

6.189 f_dividedpolygon_class::F_DividedPolygon_calculateAndSetPlaneOfExistence Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_calculateandsetplaneofexistence** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.190 f_dividedpolygon_class::F_DividedPolygon_calculateNormal Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_calculatenormal** (this, a_normal)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.191 f_dividedpolygon_class::F_DividedPolygon_calculateSign Interface Reference

Public Member Functions

- real(c_double) function **f_dividedpolygon_calculatesign** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.192 f_dividedpolygon_class::F_DividedPolygon_calculateSurfaceArea Interface Reference

Public Member Functions

- real(c_double) function **f_dividedpolygon_calculatesurfacearea** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.193 f_dividedpolygon_class::F_DividedPolygon_construct Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_construct** (this, a_npts, a_pts)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.194 f_dividedpolygon_class::F_DividedPolygon_constructFromPolygon Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_constructfrompolygon** (this, a_polygon)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.195 f_dividedpolygon_class::F_DividedPolygon_delete Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_delete** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.196 f_dividedpolygon_class::F_DividedPolygon_getBoundingPts Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.197 f_dividedpolygon_class::F_DividedPolygon_getLocalizer Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_getlocalizer** (this, a_planar_localizer)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.198 f_dividedpolygon_class::F_DividedPolygon_getNumberOfPts Interface Reference

Public Member Functions

- integer(c_int) function **f_dividedpolygon_getnumberoftpts** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.199 `f_dividedpolygon_class::F_DividedPolygon_getNumberOfSimplicesInDecomposition` Interface Reference

Public Member Functions

- integer(c_int) function `f_dividedpolygon_getnumberofsimplicesindecomposition` (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.200 `f_dividedpolygon_class::F_DividedPolygon_getPlaneOfExistence` Interface Reference

Public Member Functions

- subroutine `f_dividedpolygon_getplaneofexistence` (this, a_plane)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.201 `f_dividedpolygon_class::F_DividedPolygon_getPt` Interface Reference

Public Member Functions

- subroutine `f_dividedpolygon_getpt` (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.202 `f_dividedpolygon_class::F_DividedPolygon_getSimplexFromDecomposition` Interface Reference

Public Member Functions

- subroutine `f_dividedpolygon_getsimplexfromdecomposition` (this, a_tri_number_to_get, a_triangle_in_decomposition)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.203 f_dividedpolygon_class::F_DividedPolygon_new Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_new** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.204 f_dividedpolygon_class::F_DividedPolygon_printToScreen Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_printtoscreen** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.205 f_dividedpolygon_class::F_DividedPolygon_resetCentroid Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_resetcentroid** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.206 f_dividedpolygon_class::F_DividedPolygon_reversePtOrdering Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_reverseptordering** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.207 f_dividedpolygon_class::F_DividedPolygon_setPlaneOfExistence Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_setplaneofexistence** (this, a_plane)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.208 f_dividedpolygon_class::F_DividedPolygon_zeroPolygon Interface Reference

Public Member Functions

- subroutine **f_dividedpolygon_zeropolygon** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.209 f_dodecahedron_class::F_Dodecahedron_construct Interface Reference

Public Member Functions

- subroutine **f_dodecahedron_construct** (this, a_transported_cell)

The documentation for this interface was generated from the following file:

- [f_dodecahedron_class.f90](#)

6.210 f_dodecahedron_class::F_Dodecahedron_delete Interface Reference

Public Member Functions

- subroutine **f_dodecahedron_delete** (this)

The documentation for this interface was generated from the following file:

- [f_dodecahedron_class.f90](#)

6.211 f_dodecahedron_class::F_Dodecahedron_getBoundingPts Interface Reference

Public Member Functions

- subroutine **f_dodecahedron_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_dodecahedron_class.f90](#)

6.212 f_dodecahedron_class::F_Dodecahedron_new Interface Reference

Public Member Functions

- subroutine **f_dodecahedron_new** (this)

The documentation for this interface was generated from the following file:

- [f_dodecahedron_class.f90](#)

6.213 f_elviraneighborhood_class::F_ELVIRANeighborhood_delete Interface Reference

Public Member Functions

- subroutine **f_elviraneighborhood_delete** (this)

The documentation for this interface was generated from the following file:

- [f_elviraneighborhood_class.f90](#)

6.214 f_elviraneighborhood_class::F_ELVIRANeighborhood_new Interface Reference

Public Member Functions

- subroutine **f_elviraneighborhood_new** (this)

The documentation for this interface was generated from the following file:

- [f_elviraneighborhood_class.f90](#)

6.215 f_elviraneighborhood_class::F_ELVIRANeighborhood_setMember Interface Reference

Public Member Functions

- subroutine **f_elviraneighborhood_setmember** (this, a_rectangular_cuboid, a_liquid_volume_fraction, i, j, k)

The documentation for this interface was generated from the following file:

- [f_elviraneighborhood_class.f90](#)

6.216 f_elviraneighborhood_class::F_ELVIRANeighborhood_setSize Interface Reference

Public Member Functions

- subroutine **f_elviraneighborhood_setsize** (this, a_size)

The documentation for this interface was generated from the following file:

- [f_elviraneighborhood_class.f90](#)

6.217 f_cutpolygon::F_getPlanePolygonFromReconstruction_RC_DivPoly Interface Reference

Public Member Functions

- subroutine **f_getplanepolygonfromreconstruction_rc_divpoly** (a_rectangular_cuboid, a_planar_separator, a_plane_index, a_divided_polygon)

The documentation for this interface was generated from the following file:

- [f_cutpolygon.f90](#)

6.218 f_cutpolygon::F_getPlanePolygonFromReconstruction_RC_Poly Interface Reference

Public Member Functions

- subroutine **f_getplanepolygonfromreconstruction_rc_poly** (a_rectangular_cuboid, a_planar_separator, a_plane_index, a_polygon)

The documentation for this interface was generated from the following file:

- [f_cutpolygon.f90](#)

6.219 f_cutpolygon::F_getReconstructionSurfaceArea_RC Interface Reference

Public Member Functions

- real(c_double) function **f_getreconstructionsurfacearea_rc** (a_rectangular_cuboid, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_cutpolygon.f90](#)

6.220 f_getvolumemoments::F_GNVM_CD_By_LSL_For_SVM Interface Reference

Public Member Functions

- subroutine **f_gnvm_cd_by_lsl_for_svm** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.221 f_getvolumemoments::F_GNVM_CD_By_LSL_For_TagAccumVM_SVM Interface Reference

Public Member Functions

- subroutine **f_gnvm_cd_by_lsl_for_tagaccumvm_svm** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.222 f_getvolumemoments::F_GNVM_CDWD3_By_LSL_For_SVMAD3 Interface Reference

Public Member Functions

- subroutine **f_gnvm_cdwd3_by_lsl_for_svmad3** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.223 f_getvolumemoments::F_GNVM_D_By_LSL_For_SVM Interface Reference

Public Member Functions

- subroutine **f_gnvm_d_by_lsl_for_svm** (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.224 f_getvolumemoments::F_GNVM_D_By_LSL_For_TagAccumVM_SVM Interface Reference

Public Member Functions

- subroutine **f_gnvm_d_by_lsl_for_tagaccumvm_svm** (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.225 f_getvolumemoments::F_GNVM_D_By_PS_For_SVM Interface Reference

Public Member Functions

- subroutine **f_gnvm_d_by_ps_for_svm** (a_Dodecahedron, a_planar_separator, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.226 f_getvolumemoments::F_GNVM_P24_By_LSL_For_SVM Interface Reference

Public Member Functions

- subroutine **f_gnvm_p24_by_lsl_for_svm** (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.227 f_getvolumemoments::F_GNVM_P24WD3_By_LSL_For_SVMAD3 Interface Reference

Public Member Functions

- subroutine **f_gnvm_p24wd3_by_lsl_for_svmad3** (a_polyhedron_24, a_localized_separator_link, a_← moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.228 f_getvolumemoments::F_GNVM_Poly_By_PL_For_V Interface Reference

Public Member Functions

- subroutine **f_gnvm_poly_by_pl_for_v** (a_poly, a_planar_localizer, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.229 f_getvolumemoments::F_GNVM_RC_By_PS_For_SVM Interface Reference

Public Member Functions

- subroutine **f_gnvm_rc_by_ps_for_svm** (a_rectangular_cuboid, a_planar_separator, a_moments_to_← return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.230 f_getvolumemoments::F_GNVM_RC_By_PS_For_V Interface Reference

Public Member Functions

- subroutine **f_gnvm_rc_by_ps_for_v** (a_rectangular_cuboid, a_planar_separator, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.231 f_getvolumemoments::F_GNVM_Tet_By_LSL_For_SVM Interface Reference

Public Member Functions

- subroutine **f_gnvm_tet_by_lsl_for_svm** (a_tet, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.232 f_getvolumemoments::F_GNVM_Tri_By_LL_For_TagAVM_VM Interface Reference

Public Member Functions

- subroutine **f_gnvm_tri_by_ll_for_tagavm_vm** (a_tri, a_localizer_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.233 f_getvolumemoments::F_GNVM_Tri_By_PL_For_V Interface Reference

Public Member Functions

- subroutine **f_gnvm_tri_by_pl_for_v** (a_tri, a_planar_localizer, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.234 f_getvolumemoments::F_GVM_CD_By_LSL_For_SVM Interface Reference

Public Member Functions

- subroutine **f_gvm_cd_by_lsl_for_svm** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.235 f_getvolumemoments::F_GVM_D_By_LSL_For_SVM Interface Reference

Public Member Functions

- subroutine **f_gvm_d_by_lsl_for_svm** (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.236 f_getvolumemoments::F_GVM_P24_By_LSL_For_SVM Interface Reference

Public Member Functions

- subroutine **f_gvm_p24_by_lsl_for_svm** (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.237 f_getvolumemoments::F_GVM_setMethod Interface Reference

Public Member Functions

- subroutine **f_gvm_setmethod** (a_cutting_method)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.238 f_getvolumemoments::F_GVM_Tri_By_LL_For_TagALVM_VMAN Interface Reference

Public Member Functions

- subroutine **f_gvm_tri_by_ll_for_tagalvm_vman** (a_tri, a_localizer_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.239 f_geometriccuttinghelpers::F_isPtInternal_PL Interface Reference

Public Member Functions

- logical(c_bool) function **f_isptinternal_pl** (a_pt, a_localizer)

The documentation for this interface was generated from the following file:

- [f_geometriccuttinghelpers.f90](#)

6.240 f_geometriccuttinghelpers::F_isPtInternal_PS Interface Reference

Public Member Functions

- logical(c_bool) function **f_isptinternal_ps** (a_pt, a_separator)

The documentation for this interface was generated from the following file:

- [f_geometriccuttinghelpers.f90](#)

6.241 f_listedvm_vman_class::F_ListedVM_VMAN_append Interface Reference

Public Member Functions

- subroutine **f_listedvm_vman_append** (this, a_other_list)

The documentation for this interface was generated from the following file:

- [f_listedvm_vman_class.f90](#)

6.242 f_listedvm_vman_class::F_ListedVM_VMAN_clear Interface Reference

Public Member Functions

- subroutine **f_listedvm_vman_clear** (this)

The documentation for this interface was generated from the following file:

- [f_listedvm_vman_class.f90](#)

6.243 f_listedvm_vman_class::F_ListedVM_VMAN_delete Interface Reference

Public Member Functions

- subroutine **f_listedvm_vman_delete** (this)

The documentation for this interface was generated from the following file:

- f_listedvm_vman_class.f90

6.244 f_listedvm_vman_class::F_ListedVM_VMAN_erase Interface Reference

Public Member Functions

- subroutine **f_listedvm_vman_erase** (this, a_index)

The documentation for this interface was generated from the following file:

- f_listedvm_vman_class.f90

6.245 f_listedvm_vman_class::F_ListedVM_VMAN_getMoments Interface Reference

Public Member Functions

- subroutine **f_listedvm_vman_getmoments** (this, a_index, a_moments)

The documentation for this interface was generated from the following file:

- f_listedvm_vman_class.f90

6.246 f_listedvm_vman_class::F_ListedVM_VMAN_getSize Interface Reference

Public Member Functions

- integer(c_int) function **f_listedvm_vman_getsize** (this)

The documentation for this interface was generated from the following file:

- f_listedvm_vman_class.f90

6.247 `f_listedvm_vman_class::F_ListedVM_VMAN_new` Interface Reference

Public Member Functions

- subroutine `f_listedvm_vman_new` (`this`)

The documentation for this interface was generated from the following file:

- `f_listedvm_vman_class.f90`

6.248 `f_listedvm_vman_class::F_ListedVM_VMAN_zeroNormalComponent` Interface Reference

Public Member Functions

- subroutine `f_listedvm_vman_zeronormalcomponent` (`this, a_index`)

The documentation for this interface was generated from the following file:

- `f_listedvm_vman_class.f90`

6.249 `f_localizedseparatorlink_class::F_LocalizedSeparatorLink_delete` Interface Reference

Public Member Functions

- subroutine `f_localizedseparatorlink_delete` (`this`)

The documentation for this interface was generated from the following file:

- `f_localizedseparatorlink_class.f90`

6.250 `f_localizedseparatorlink_class::F_LocalizedSeparatorLink_getId` Interface Reference

Public Member Functions

- integer(`c_int`) function `f_localizedseparatorlink_getid` (`this`)

The documentation for this interface was generated from the following file:

- `f_localizedseparatorlink_class.f90`

6.251 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_new Interface Reference

Public Member Functions

- subroutine **f_localizedseparatorlink_new** (this, a_planar_localizer, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.252 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_newFromObject← AllocationServer Interface Reference

Public Member Functions

- subroutine **f_localizedseparatorlink_newfromobjectallocationserver** (this, a_object_allocation_server, a_planar_localizer, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.253 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivity Interface Reference

Public Member Functions

- subroutine **f_localizedseparatorlink_setedgeconnectivity** (this, a_plane_index, a_ptr_to_neighbor)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.254 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivity← Null Interface Reference

Public Member Functions

- subroutine **f_localizedseparatorlink_setedgeconnectivitynull** (this, a_plane_index)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.255 f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setId Interface Reference

Public Member Functions

- subroutine **f_localizedseparatorlink_setid** (this, a_id)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.256 f_localizerlink_class::F_LocalizerLink_delete Interface Reference

Public Member Functions

- subroutine **f_localizerlink_delete** (this)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.257 f_localizerlink_class::F_LocalizerLink_getId Interface Reference

Public Member Functions

- integer(c_int) function **f_localizerlink_getid** (this)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.258 f_localizerlink_class::F_LocalizerLink_new Interface Reference

Public Member Functions

- subroutine **f_localizerlink_new** (this, a_planar_localizer)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.259 f_localizerlink_class::F_LocalizerLink_newFromObjectAllocationServer Interface Reference

Public Member Functions

- subroutine **f_localizerlink_newfromobjectallocationserver** (this, a_object_allocation_server, a_planar← localizer)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.260 f_localizerlink_class::F_LocalizerLink_setEdgeConnectivity Interface Reference

Public Member Functions

- subroutine **f_localizerlink_setedgeconnectivity** (this, a_plane_index, a_ptr_to_neighbor)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.261 f_localizerlink_class::F_LocalizerLink_setEdgeConnectivityNull Interface Reference

Public Member Functions

- subroutine **f_localizerlink_setedgeconnectivitynull** (this, a_plane_index)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.262 f_localizerlink_class::F_LocalizerLink_setId Interface Reference

Public Member Functions

- subroutine **f_localizerlink_setid** (this, a_id)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.263

f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_addMember

Interface Reference

233

6.263 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_addMember Interface Reference

Public Member Functions

- subroutine **f_lviraneighborhood_rectangularcuboid_addmember** (this, a_rectangular_cuboid, a_<volume_fraction>)

The documentation for this interface was generated from the following file:

- f_lviraneighborhood_rectangularcuboid_class.f90

6.264 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_delete Interface Reference

Public Member Functions

- subroutine **f_lviraneighborhood_rectangularcuboid_delete** (this)

The documentation for this interface was generated from the following file:

- f_lviraneighborhood_rectangularcuboid_class.f90

6.265 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_emptyNeighborhood Interface Reference

Public Member Functions

- subroutine **f_lviraneighborhood_rectangularcuboid_emptyneighborhood** (this)

The documentation for this interface was generated from the following file:

- f_lviraneighborhood_rectangularcuboid_class.f90

6.266 f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_RectangularCuboid_new Interface Reference

Public Member Functions

- subroutine **f_lviraneighborhood_rectangularcuboid_new** (this)

The documentation for this interface was generated from the following file:

- f_lviraneighborhood_rectangularcuboid_class.f90

6.267 `f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_Rectangular←Cuboid_setCenterOfStencil` Interface Reference

Public Member Functions

- subroutine `f_lviraneighborhood_rectangularcuboid_setcenterofstencil` (this, a_center_cell_index)

The documentation for this interface was generated from the following file:

- `f_lviraneighborhood_rectangularcuboid_class.f90`

6.268 `f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_Rectangular←Cuboid_setMember` Interface Reference

Public Member Functions

- subroutine `f_lviraneighborhood_rectangularcuboid_setmember` (this, a_index, a_rectangular_cuboid, a_liquid_volume_fraction)

The documentation for this interface was generated from the following file:

- `f_lviraneighborhood_rectangularcuboid_class.f90`

6.269 `f_lviraneighborhood_rectangularcuboid_class::F_LVIRANeighborhood_Rectangular←Cuboid_setSize` Interface Reference

Public Member Functions

- subroutine `f_lviraneighborhood_rectangularcuboid_setsize` (this, a_size)

The documentation for this interface was generated from the following file:

- `f_lviraneighborhood_rectangularcuboid_class.f90`

6.270 `f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer←_LocalizedSeparatorLink_delete` Interface Reference

Public Member Functions

- subroutine `f_objectallocationserver_localizedseparatorlink_delete` (this)

The documentation for this interface was generated from the following file:

- `f_objectallocationserver_localizedseparatorlink_class.f90`

6.271 f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparatorLink_new Interface

Reference

235

6.271 f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparatorLink_new Interface Reference

Public Member Functions

- subroutine **f_objectallocationserver_localizedseparatorlink_new** (this, a_number_to_allocate)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_localizedseparatorlink_class.f90](#)

6.272 f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_delete Interface Reference

Public Member Functions

- subroutine **f_objectallocationserver_localizerlink_delete** (this)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_localizerlink_class.f90](#)

6.273 f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_new Interface Reference

Public Member Functions

- subroutine **f_objectallocationserver_localizerlink_new** (this, a_number_to_allocate)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_localizerlink_class.f90](#)

6.274 f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_delete Interface Reference

Public Member Functions

- subroutine **f_objectallocationserver_planarlocalizer_delete** (this)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_planarlocalizer_class.f90](#)

6.275 f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_← PlanarLocalizer_new Interface Reference

Public Member Functions

- subroutine **f_objectallocationserver_planarlocalizer_new** (this, a_number_to_allocate)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_planarlocalizer_class.f90](#)

6.276 f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_← PlanarSeparator_delete Interface Reference

Public Member Functions

- subroutine **f_objectallocationserver_planarseparator_delete** (this)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_planarseparator_class.f90](#)

6.277 f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_← PlanarSeparator_new Interface Reference

Public Member Functions

- subroutine **f_objectallocationserver_planarseparator_new** (this, a_number_to_allocate)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_planarseparator_class.f90](#)

6.278 f_planarlocalizer_class::F_PlanarLocalizer_addPlane Interface Reference

Public Member Functions

- subroutine **f_planarlocalizer_addplane** (this, a_normal, a_distance)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.279 f_planarlocalizer_class::F_PlanarLocalizer_delete Interface Reference

Public Member Functions

- subroutine **f_planarlocalizer_delete** (this)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.280 f_planarlocalizer_class::F_PlanarLocalizer_new Interface Reference

Public Member Functions

- subroutine **f_planarlocalizer_new** (this)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.281 f_planarlocalizer_class::F_PlanarLocalizer_newFromObjectAllocationServer Interface Reference

Public Member Functions

- subroutine **f_planarlocalizer_newfromobjectallocationserver** (this, a_object_allocation_server)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.282 f_planarlocalizer_class::F_PlanarLocalizer_printToScreen Interface Reference

Public Member Functions

- subroutine **f_planarlocalizer_printtoscreen** (this)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.283 f_planarlocalizer_class::F_PlanarLocalizer_setFromRectangularCuboid Interface Reference

Public Member Functions

- subroutine **f_planarlocalizer_setfromrectangularcuboid** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.284 f_planarlocalizer_class::F_PlanarLocalizer_setNumberOfPlanes Interface Reference

Public Member Functions

- subroutine **f_planarlocalizer_setnumberofplanes** (this, a_number_to_set)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.285 f_planarlocalizer_class::F_PlanarLocalizer_setPlane Interface Reference

Public Member Functions

- subroutine **f_planarlocalizer_setplane** (this, a_plane_index_to_set, a_normal, a_distance)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.286 f_planarseparator_class::F_PlanarSeparator_addPlane Interface Reference

Public Member Functions

- subroutine **f_planarseparator_addplane** (this, a_normal, a_distance)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.287 f_planarseparator_class::F_PlanarSeparator_copy Interface Reference

Public Member Functions

- subroutine **f_planarseparator_copy** (this, a_other_PlanarSeparator)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.288 f_planarseparator_class::F_PlanarSeparator_delete Interface Reference

Public Member Functions

- subroutine **f_planarseparator_delete** (this)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.289 f_planarseparator_class::F_PlanarSeparator_getNumberOfPlanes Interface Reference

Public Member Functions

- integer(c_int) function **f_planarseparator_getnumberofplanes** (this)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.290 f_planarseparator_class::F_PlanarSeparator_getPlane Interface Reference

Public Member Functions

- subroutine **f_planarseparator_getplane** (this, a_index, a_plane_listed)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.291 f_planarseparator_class::F_PlanarSeparator_isFlipped Interface Reference

Public Member Functions

- logical(c_bool) function **f_planarseparator_isflipped** (this)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.292 f_planarseparator_class::F_PlanarSeparator_new Interface Reference

Public Member Functions

- subroutine **f_planarseparator_new** (this)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.293 f_planarseparator_class::F_PlanarSeparator_newFromObjectAllocationServer Interface Reference

Public Member Functions

- subroutine **f_planarseparator_newfromobjectallocationserver** (this, a_object_allocation_server)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.294 f_planarseparator_class::F_PlanarSeparator_printToScreen Interface Reference

Public Member Functions

- subroutine **f_planarseparator_printtoscreen** (this)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.295 f_planarseparator_class::F_PlanarSeparator_setNumberOfPlanes Interface Reference

Public Member Functions

- subroutine **f_planarseparator_setnumberofplanes** (this, a_number_to_set)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.296 f_planarseparator_class::F_PlanarSeparator_setPlane Interface Reference

Public Member Functions

- subroutine **f_planarseparator_setplane** (this, a_plane_index_to_set, a_normal, a_distance)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.297 f_polygon_class::F_Polygon_calculateAndSetPlaneOfExistence Interface Reference

Public Member Functions

- subroutine **f_polygon_calculateandsetplaneofexistence** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.298 f_polygon_class::F_Polygon_calculateCentroid Interface Reference

Public Member Functions

- subroutine **f_polygon_calculatecentroid** (this, a_centroid)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.299 f_polygon_class::F_Polygon_calculateNearestPtOnSurface Interface Reference

Public Member Functions

- subroutine **f_polygon_calculatenearstptonsurface** (this, a_pt, a_pt_on_polygon)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.300 f_polygon_class::F_Polygon_calculateNormal Interface Reference

Public Member Functions

- subroutine **f_polygon_calculatenormal** (this, a_normal)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.301 f_polygon_class::F_Polygon_calculateSign Interface Reference

Public Member Functions

- real(c_double) function **f_polygon_calculatesign** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.302 f_polygon_class::F_Polygon_calculateVolume Interface Reference

Public Member Functions

- real(c_double) function **f_polygon_calculatevolume** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.303 f_polygon_class::F_Polygon_construct Interface Reference

Public Member Functions

- subroutine **f_polygon_construct** (this, a_npts, a_pts)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.304 f_polygon_class::F_Polygon_delete Interface Reference

Public Member Functions

- subroutine **f_polygon_delete** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.305 f_polygon_class::F_Polygon_getBoundingPts Interface Reference

Public Member Functions

- subroutine **f_polygon_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.306 f_polygon_class::F_Polygon_getLocalizer Interface Reference

Public Member Functions

- subroutine **f_polygon_getlocalizer** (this, a_planar_localizer)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.307 f_polygon_class::F_Polygon_getNumberOfPts Interface Reference

Public Member Functions

- integer(c_int) function **f_polygon_getnumberoftpts** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.308 f_polygon_class::F_Polygon_getNumberOfSimplicesInDecomposition Interface Reference

Public Member Functions

- integer(c_int) function **f_polygon_getnumberofsimplicesindecomposition** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.309 f_polygon_class::F_Polygon_getPlaneOfExistence Interface Reference

Public Member Functions

- subroutine **f_polygon_getplaneofexistence** (this, a_plane)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.310 f_polygon_class::F_Polygon_getPt Interface Reference

Public Member Functions

- subroutine **f_polygon_getpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.311 f_polygon_class::F_Polygon_getSimplexFromDecomposition Interface Reference

Public Member Functions

- subroutine **f_polygon_getsimplexfromdecomposition** (this, a_tri_number_to_get, a_triangle_in_← decomposition)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.312 f_polygon_class::F_Polygon_new Interface Reference

Public Member Functions

- subroutine **f_polygon_new** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.313 f_polygon_class::F_Polygon_printToScreen Interface Reference

Public Member Functions

- subroutine **f_polygon_printtoscreen** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.314 f_polygon_class::F_Polygon_reversePtOrdering Interface Reference

Public Member Functions

- subroutine **f_polygon_reverseptordering** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.315 f_polygon_class::F_Polygon_setPlaneOfExistence Interface Reference

Public Member Functions

- subroutine **f_polygon_setplaneofexistence** (this, a_plane)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.316 f_polygon_class::F_Polygon_zeroPolygon Interface Reference

Public Member Functions

- subroutine **f_polygon_zeropolygon** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.317 f_polyhedron24_class::F_Polyhedron24_adjustCapToMatchVolume Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_adjustcaptomatchvolume** (this, a_correct_signed_volume)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.318 f_polyhedron24_class::F_Polyhedron24_construct Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_construct** (this, a_polyhedron24)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.319 f_polyhedron24_class::F_Polyhedron24_delete Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_delete** (this)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.320 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_adjustCapToMatchVolume Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_doubles3_adjustcaptomatchvolume** (this, a_correct_signed_volume)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.321 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_construct Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_doubles3_construct** (this, a_polyhedron24, a_data)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.322 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_delete Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_doubles3_delete** (this)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.323 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getBoundingPts Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_doubles3_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.324 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getData Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_doubles3_getdata** (this, a_index, a_data)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.325 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getPt Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_doubles3_getpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.326 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_new Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_doubles3_new** (this)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.327 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setData Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_doubles3_setdata** (this, a_index, a_data)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.328 f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setPt Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_doubles3_setpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.329 f_polyhedron24_class::F_Polyhedron24_getBoundingPts Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.330 f_polyhedron24_class::F_Polyhedron24_getPt Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_getpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.331 f_polyhedron24_class::F_Polyhedron24_new Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_new** (this)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.332 f_polyhedron24_class::F_Polyhedron24_setPt Interface Reference

Public Member Functions

- subroutine **f_polyhedron24_setpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.333 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_Rectangular← Cuboid_addMember Interface Reference

Public Member Functions

- subroutine **f_r2pneighborhood_rectangularcuboid_addmember** (this, a_rectangular_cuboid, a_separated_volume_moments)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.334 f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_Rectangular← Cuboid_delete Interface Reference

Public Member Functions

- subroutine **f_r2pneighborhood_rectangularcuboid_delete** (this)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.335 ~~f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_empty~~
Neighborhood Interface

Reference

6.335 ~~f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_Rectangular~~
Cuboid_empty Neighborhood Interface Reference 251

Public Member Functions

- subroutine **f_r2pneighborhood_rectangularcuboid_emptyneighborhood** (this)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.336 ~~f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_Rectangular~~
Cuboid_new Interface Reference

Public Member Functions

- subroutine **f_r2pneighborhood_rectangularcuboid_new** (this)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.337 ~~f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_Rectangular~~
Cuboid_setCenterOfStencil Interface Reference

Public Member Functions

- subroutine **f_r2pneighborhood_rectangularcuboid_setcenterofstencil** (this, a_center_cell_index)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.338 ~~f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_Rectangular~~
Cuboid_setMember Interface Reference

Public Member Functions

- subroutine **f_r2pneighborhood_rectangularcuboid_setmember** (this, a_rectangular_cuboid, a_separated_volume_moments, a_index)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.339 `f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setSize` Interface Reference

Public Member Functions

- subroutine `f_r2pneighborhood_rectangularcuboid_setsize` (this, a_size)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.340 `f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setSurfaceArea` Interface Reference

Public Member Functions

- subroutine `f_r2pneighborhood_rectangularcuboid_setsurfacearea` (this, a_surface_area)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.341 `f_reconstructioninterface::F_reconstructionWithAdvectedNormals_ListedVM_VMAN_RC` Interface Reference

Public Member Functions

- subroutine `f_reconstructionwithadvectednormals_listedvm_vman_rc` (a_volume_moments_list, a_neighborhood, a_two_plane_threshold, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_reconstructioninterface.f90](#)

6.342 `f_reconstructioninterface::F_reconstructionWithAdvectedNormalsDebug_ListedVM_VMAN_RC` Interface Reference

Public Member Functions

- subroutine `f_reconstructionwithadvectednormalsdebug_listedvm_vman_rc` (a_volume_moments_list, a_neighborhood, a_two_plane_threshold, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_reconstructioninterface.f90](#)

6.343 f_reconstructioninterface::F_reconstructionWithELVIRA2D Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithelvira2d** (a_ELVIRANeighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.344 f_reconstructioninterface::F_reconstructionWithELVIRA3D Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithelvira3d** (a_ELVIRANeighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.345 f_reconstructioninterface::F_reconstructionWithLVIRA2D_RC Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithlvira2d_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.346 f_reconstructioninterface::F_reconstructionWithLVIRA3D_RC Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithlvira3d_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.347 f_reconstructioninterface::F_reconstructionWithMOF2D_RectangularCuboid Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithmof2d_rectangularcuboid** (a_rectangular_cuboid, a_separated_← volume_moments, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.348 f_reconstructioninterface::F_reconstructionWithMOF2D_Tri Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithmof2d_tri** (a_tri, a_separated_volume_moments, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.349 f_reconstructioninterface::F_reconstructionWithMOF2DGiveWeights_Rectangular_← Cuboid Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithmof2dgiveweights_rectangularcuboid** (a_rectangular_cuboid, a_← separated_volume_moments, a_internal_weight, a_external_weight, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.350 f_reconstructioninterface::F_reconstructionWithMOF2DGiveWeights_Tri Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithmof2dgiveweights_tri** (a_tri, a_separated_volume_moments, a_← internal_weight, a_external_weight, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.351 f_reconstructioninterface::F_reconstructionWithMOF3D_RectangularCuboid Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithmof3d_rectangularcuboid** (a_rectangular_cuboid, a_separated_← volume_moments, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.352 f_reconstructioninterface::F_reconstructionWithMOF3D_Tet Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithmof3d_tet** (a_tet, a_separated_volume_moments, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.353 f_reconstructioninterface::F_reconstructionWithMOF3DGiveWeights_Rectangular_← Cuboid Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithmof3dgiveweights_rectangularcuboid** (a_rectangular_cuboid, a_← separated_volume_moments, a_internal_weight, a_external_weight, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.354 f_reconstructioninterface::F_reconstructionWithMOF3DGiveWeights_Tet Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithmof3dgiveweights_tet** (a_tet, a_separated_volume_moments, a_← internal_weight, a_external_weight, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.355 f_reconstructioninterface::F_reconstructionWithR2P2D_RC Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithr2p2d_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.356 f_reconstructioninterface::F_reconstructionWithR2P2DDebug_RC Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithr2p2ddebug_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.357 f_reconstructioninterface::F_reconstructionWithR2P3D_RC Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithr2p3d_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.358 f_reconstructioninterface::F_reconstructionWithR2P3DDebug_RC Interface Reference

Public Member Functions

- subroutine **f_reconstructionwithr2p3ddebug_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.359 f_rectangularcuboid_class::F_RectangularCuboid_calculateVolume Interface Reference

Public Member Functions

- real(c_double) function **f_rectangularcuboid_calculatevolume** (this)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.360 f_rectangularcuboid_class::F_RectangularCuboid_construct Interface Reference

Public Member Functions

- subroutine **f_rectangularcuboid_construct** (this, a_transported_cell)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.361 f_rectangularcuboid_class::F_RectangularCuboid_construct_2pt Interface Reference

Public Member Functions

- subroutine **f_rectangularcuboid_construct_2pt** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.362 f_rectangularcuboid_class::F_RectangularCuboid_delete Interface Reference

Public Member Functions

- subroutine **f_rectangularcuboid_delete** (this)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.363 f_rectangularcuboid_class::F_RectangularCuboid_getBoundingPts Interface Reference

Public Member Functions

- subroutine **f_rectangularcuboid_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.364 f_rectangularcuboid_class::F_RectangularCuboid_new Interface Reference

Public Member Functions

- subroutine **f_rectangularcuboid_new** (this)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.365 f_sepvm_class::F_SepVM_construct Interface Reference

Public Member Functions

- subroutine **f_sepvm_construct** (this, a_moments_list)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.366 f_sepvm_class::F_SepVM_delete Interface Reference

Public Member Functions

- subroutine **f_sepvm_delete** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.367 f_sepvm_doubles3_class::F_SepVM_doubles3_delete Interface Reference

Public Member Functions

- subroutine **f_sepvm_doubles3_delete** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.368 f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroid Interface Reference

Public Member Functions

- subroutine **f_sepvm_doubles3_getcentroid** (this, a_index, a_centroid)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.369 f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroidPtr Interface Reference

Public Member Functions

- type(c_ptr) function **f_sepvm_doubles3_getcentroidptr** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.370 f_sepvm_doubles3_class::F_SepVM_doubles3_getData Interface Reference

Public Member Functions

- subroutine **f_sepvm_doubles3_getdata** (this, a_index, a_data)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.371 f_sepvm_doubles3_class::F_SepVM_doubles3_getVolume Interface Reference

Public Member Functions

- real(c_double) function **f_sepvm_doubles3_getvolume** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.372 f_sepvm_doubles3_class::F_SepVM_doubles3_getVolumePtr Interface Reference

Public Member Functions

- type(c_ptr) function **f_sepvm_doubles3_getvolumeptr** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.373 f_sepvm_doubles3_class::F_SepVM_doubles3_multiplyByVolume Interface Reference

Public Member Functions

- subroutine **f_sepvm_doubles3_multiplybyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.374 f_sepvm_doubles3_class::F_SepVM_doubles3_new Interface Reference

Public Member Functions

- subroutine **f_sepvm_doubles3_new** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.375 f_sepvm_doubles3_class::F_SepVM_doubles3_normalizeByVolume Interface Reference

Public Member Functions

- subroutine **f_sepvm_doubles3_normalizebyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.376 f_sepvm_class::F_SepVM_getCentroid Interface Reference

Public Member Functions

- subroutine **f_sepvm_getcentroid** (this, a_index, a_centroid)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.377 f_sepvm_class::F_SepVM_getCentroidPtr Interface Reference

Public Member Functions

- type(c_ptr) function **f_sepvm_getcentroidptr** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.378 f_sepvm_class::F_SepVM_getVolume Interface Reference

Public Member Functions

- real(c_double) function **f_sepvm_getvolume** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.379 f_sepvm_class::F_SepVM_getVolumePtr Interface Reference

Public Member Functions

- type(c_ptr) function **f_sepvm_getvolumeptr** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.380 f_sepvm_class::F_SepVM_multiplyByVolume Interface Reference

Public Member Functions

- subroutine **f_sepvm_multiplybyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.381 f_sepvm_class::F_SepVM_new Interface Reference

Public Member Functions

- subroutine **f_sepvm_new** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.382 f_sepvm_class::F_SepVM_normalizeByVolume Interface Reference

Public Member Functions

- subroutine **f_sepvm_normalizebyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.383 f_serializer::F_Serializer_serializeAndPack_PlanarSeparator_ByteBuffer Interface Reference

Public Member Functions

- subroutine **f_serializer_serializeandpack_planarseparator_bytebuffer** (a_separator, a_byte_buffer)

The documentation for this interface was generated from the following file:

- [f_serializer.f90](#)

6.384 f_serializer::F_Serializer_unpackAndStore_PlanarSeparator_ByteBuffer Interface Reference

Public Member Functions

- subroutine **f_serializer_unpackandstore_planarseparator_bytebuffer** (a_separator, a_byte_buffer)

The documentation for this interface was generated from the following file:

- [f_serializer.f90](#)

6.385 f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS Interface Reference

Public Member Functions

- subroutine **f_setdistancetomatchvolumefraction_rc_ps** (a_rectangular_cuboid, a_volume_fraction, a_planar_separator, a_volume_fraction_tolerance)

The documentation for this interface was generated from the following file:

- [f_volumefractionmatching.f90](#)

6.386 f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS_DefTol Interface Reference

Public Member Functions

- subroutine **f_setdistancetomatchvolumefraction_rc_ps_deftol** (a_rectangular_cuboid, a_volume_fraction, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_volumefractionmatching.f90](#)

6.387 `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_append` Interface Reference

Public Member Functions

- subroutine `f_tagged_accumlistedvm_vman_append` (this, a_other_list)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.388 `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_clear` Interface Reference

Public Member Functions

- subroutine `f_tagged_accumlistedvm_vman_clear` (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.389 `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_delete` Interface Reference

Public Member Functions

- subroutine `f_tagged_accumlistedvm_vman_delete` (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.390 `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getAtIndex` Interface Reference

Public Member Functions

- subroutine `f_tagged_accumlistedvm_vman_getlistatindex` (this, a_index, a_other_list)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.391 `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getSize` Interface Reference

Public Member Functions

- integer(c_int) function `f_tagged_accumlistedvm_vman_getsize` (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.392 `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getTagForIndex` Interface Reference

Public Member Functions

- integer(c_int) function `f_tagged_accumlistedvm_vman_gettagforindex` (this, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.393 `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_new` Interface Reference

Public Member Functions

- subroutine `f_tagged_accumlistedvm_vman_new` (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.394 `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_delete` Interface Reference

Public Member Functions

- subroutine `f_tagged_accumvm_sepvm_delete` (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.395 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtIndex Interface Reference

Public Member Functions

- subroutine **f_tagged_accumvm_sepvm_getcentroidatindex** (this, a_list_index, a_index, a_centroid)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.396 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtTag Interface Reference

Public Member Functions

- subroutine **f_tagged_accumvm_sepvm_getcentroidattag** (this, a_tag, a_index, a_centroid)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.397 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidPtrAtIndex Interface Reference

Public Member Functions

- type(c_ptr) function **f_tagged_accumvm_sepvm_getcentroidptratindex** (this, a_list_index, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.398 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getSize Interface Reference

Public Member Functions

- integer(c_int) function **f_tagged_accumvm_sepvm_getsize** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.399 `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getTagForIndex` Interface Reference

Public Member Functions

- integer(c_int) function `f_tagged_accumvm_sepvm_gettagforindex` (this, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.400 `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtIndex` Interface Reference

Public Member Functions

- real(c_double) function `f_tagged_accumvm_sepvm_getvolumeatindex` (this, a_list_index, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.401 `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtTag` Interface Reference

Public Member Functions

- real(c_double) function `f_tagged_accumvm_sepvm_getvolumeattag` (this, a_tag, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.402 `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumePtrAtIndex` Interface Reference

Public Member Functions

- type(c_ptr) function `f_tagged_accumvm_sepvm_getvolumeptratindex` (this, a_list_index, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.403 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_multiplyBy← Volume Interface Reference

Public Member Functions

- subroutine **f_tagged_accumvm_sepvm_multiplybyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.404 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_new Interface Reference

Public Member Functions

- subroutine **f_tagged_accumvm_sepvm_new** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.405 f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_normalizeBy← Volume Interface Reference

Public Member Functions

- subroutine **f_tagged_accumvm_sepvm_normalizebyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.406 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_delete Interface Reference

Public Member Functions

- subroutine **f_tagged_accumvm_vm_delete** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.407 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidAtIndex Interface Reference

Public Member Functions

- subroutine **f_tagged_accumvm_vm_getcentroidatindex** (this, a_list_index, a_centroid)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.408 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidPtrAtIndex Interface Reference

Public Member Functions

- type(c_ptr) function **f_tagged_accumvm_vm_getcentroidptratindex** (this, a_list_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.409 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getSize Interface Reference

Public Member Functions

- integer(c_int) function **f_tagged_accumvm_vm_getsize** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.410 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getTagForIndex Interface Reference

Public Member Functions

- integer(c_int) function **f_tagged_accumvm_vm_gettagforindex** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.411 `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumeAtIndex` Interface Reference

Public Member Functions

- `real(c_double)` function `f_tagged_accumvm_vm_getvolumeatindex` (`this, a_list_index`)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.412 `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumePtrAtIndex` Interface Reference

Public Member Functions

- `type(c_ptr)` function `f_tagged_accumvm_vm_getvolumepratindex` (`this, a_list_index`)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.413 `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_multiplyByVolume` Interface Reference

Public Member Functions

- subroutine `f_tagged_accumvm_vm_multiplybyvolume` (`this`)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.414 `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_new` Interface Reference

Public Member Functions

- subroutine `f_tagged_accumvm_vm_new` (`this`)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.415 f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_normalizeByVolume Interface Reference

Public Member Functions

- subroutine **f_tagged_accumvm_vm_normalizebyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.416 f_tet_class::F_Tet_construct Interface Reference

Public Member Functions

- subroutine **f_tet_construct** (this, a_Tet_pts)

The documentation for this interface was generated from the following file:

- [f_tet_class.f90](#)

6.417 f_tet_class::F_Tet_delete Interface Reference

Public Member Functions

- subroutine **f_tet_delete** (this)

The documentation for this interface was generated from the following file:

- [f_tet_class.f90](#)

6.418 f_tet_class::F_Tet_getBoundingPts Interface Reference

Public Member Functions

- subroutine **f_tet_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_tet_class.f90](#)

6.419 f_tet_class::F_Tet_new Interface Reference

Public Member Functions

- subroutine **f_tet_new** (this)

The documentation for this interface was generated from the following file:

- [f_tet_class.f90](#)

6.420 f_tri_class::F_Tri_calculateAndSetPlaneOfExistence Interface Reference

Public Member Functions

- subroutine **f_tri_calculateandsetplaneofexistence** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.421 f_tri_class::F_Tri_calculateCentroid Interface Reference

Public Member Functions

- subroutine **f_tri_calculatecentroid** (this, a_centroid)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.422 f_tri_class::F_Tri_calculateNormal Interface Reference

Public Member Functions

- subroutine **f_tri_calculatenormal** (this, a_normal)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.423 f_tri_class::F_Tri_calculateSign Interface Reference

Public Member Functions

- real(c_double) function **f_tri_calculatesign** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.424 f_tri_class::F_Tri_calculateVolume Interface Reference

Public Member Functions

- real(c_double) function **f_tri_calculatevolume** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.425 f_tri_class::F_Tri_construct Interface Reference

Public Member Functions

- subroutine **f_tri_construct** (this, a_pts)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.426 f_tri_class::F_Tri_delete Interface Reference

Public Member Functions

- subroutine **f_tri_delete** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.427 f_tri_class::F_Tri_getBoundingPts Interface Reference

Public Member Functions

- subroutine **f_tri_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.428 f_tri_class::F_Tri_getLocalizer Interface Reference

Public Member Functions

- subroutine **f_tri_getlocalizer** (this, a_planar_localizer)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.429 f_tri_class::F_Tri_getPlaneOfExistence Interface Reference

Public Member Functions

- subroutine **f_tri_getplaneofexistence** (this, a_plane)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.430 f_tri_class::F_Tri_getVertices Interface Reference

Public Member Functions

- subroutine **f_tri_getvertices** (this, a_pts)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.431 f_tri_class::F_Tri_new Interface Reference

Public Member Functions

- subroutine **f_tri_new** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.432 f_tri_class::F_Tri_reversePtOrdering Interface Reference

Public Member Functions

- subroutine **f_tri_reverseptordering** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.433 f_tri_class::F_Tri_setPlaneOfExistence Interface Reference

Public Member Functions

- subroutine **f_tri_setplaneofexistence** (this, a_plane)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.434 f_vman_class::F_VMAN_delete Interface Reference

Public Member Functions

- subroutine **f_vman_delete** (this)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.435 f_vman_class::F_VMAN_getCentroid Interface Reference

Public Member Functions

- subroutine **f_vman_getcentroid** (this, a_centroid)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.436 f_vman_class::F_VMAN_getNormal Interface Reference

Public Member Functions

- subroutine **f_vman_getnormal** (this, a_normal)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.437 f_vman_class::F_VMAN_getVolume Interface Reference

Public Member Functions

- subroutine **f_vman_getvolume** (this, a_volume)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.438 f_vman_class::F_VMAN_multiplyByVolume Interface Reference

Public Member Functions

- subroutine **f_vman_multiplybyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.439 `f_vman_class::F_VMAN_new` Interface Reference

Public Member Functions

- subroutine `f_vman_new` (`this`)

The documentation for this interface was generated from the following file:

- `f_vman_class.f90`

6.440 `f_vman_class::F_VMAN_normalizeByVolume` Interface Reference

Public Member Functions

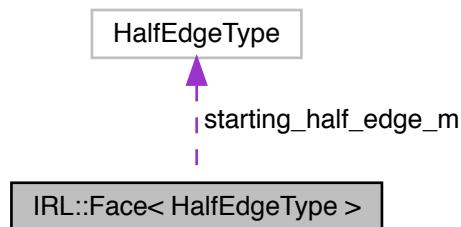
- subroutine `f_vman_normalizebyvolume` (`this`)

The documentation for this interface was generated from the following file:

- `f_vman_class.f90`

6.441 `IRL::Face< HalfEdgeType >` Class Template Reference

Collaboration diagram for `IRL::Face< HalfEdgeType >`:



Public Types

- using `half_edge_type` = `HalfEdgeType`

Public Member Functions

- **Face** (HalfEdgeType *a_starting_half_edge)
- void **setStartingHalfEdge** (HalfEdgeType *a_starting_half_edge)
- HalfEdgeType * **getStartingHalfEdge** (void)
- const HalfEdgeType * **getStartingHalfEdge** (void) const
- void **reverseFaceOrientation** (void)
- void **markAsVisited** (void)
- void **markAsNotVisited** (void)
- bool **hasBeenVisited** (void) const
- bool **hasNotBeenCalled** (void) const
- bool **checkValidFace** (void)

Static Public Member Functions

- static bool **checkValidFaceLoop** (const Face *a_face, const HalfEdgeType *a_starting_half_edge)
- static bool **checkValidCircuitOnFace** (const HalfEdgeType *a_starting_half_edge)
- static bool **checkCorrectReflectionOfFaceHalfEdgeOpposites** (const HalfEdgeType *a_starting_half_← edge)
- static bool **checkAllHalfEdgesPointBackToFace** (const Face *a_face, const HalfEdgeType *a_starting_← half_edge)

Private Attributes

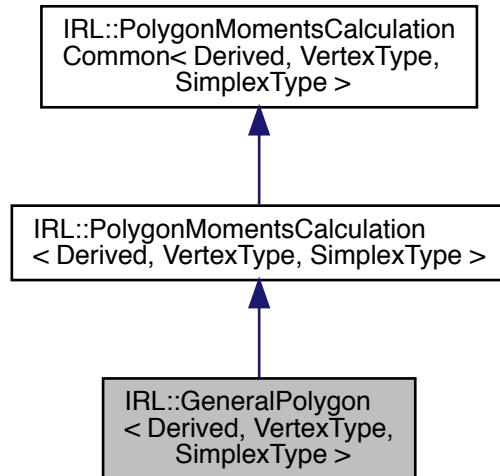
- HalfEdgeType * **starting_half_edge_m**
- bool **has_been_visited_m**

The documentation for this class was generated from the following file:

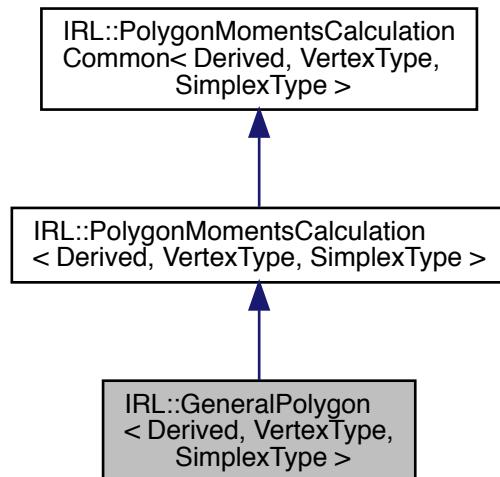
- half_edge.h

6.442 IRL::GeneralPolygon< Derived, VertexType, SimplexType > Class Template Reference

Inheritance diagram for IRL::GeneralPolygon< Derived, VertexType, SimplexType >:



Collaboration diagram for IRL::GeneralPolygon< Derived, VertexType, SimplexType >:



Public Types

- using **pt_type** = VertexType
- using **value_t** = pt_type

Public Member Functions

- VertexType & **operator[]** (const UnsignedIndex_t a_index)
- const VertexType & **operator[]** (const UnsignedIndex_t a_index) const
- UnsignedIndex_t **getNumberOfSimplicesInDecomposition** (void) const
- std::array< UnsignedIndex_t, 3 > **getSimplexIndicesFromDecomposition** (const UnsignedIndex_t a_tet) const
- SimplexType **getSimplexFromDecomposition** (const UnsignedIndex_t a_tet_number_to_get) const
- UnsignedIndex_t **getNumberOfVertices** (void) const
- void **setNumberOfVertices** (const UnsignedIndex_t a_size)
- IRL::Pt **getLowerLimits** (void) const

Return a point for the lower limits of the polyhedron in 3D space.
- IRL::Pt **getUpperLimits** (void) const

Return a point for the upper limits of the polyhedron in 3D space.
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept
- void **calculateAndSetPlaneOfExistence** (void)
- void **setPlaneOfExistence** (const Plane &a_plane)
- const Plane & **getPlaneOfExistence** (void) const
- double **calculateSign** (void) const
- Normal **calculateNormal** (void) const

Const version of calculateNormal.
- void **reversePtOrdering** (void)

Reverse point ordering.
- PlanarLocalizer **getLocalizer** (void) const

Returns a planar reconstruction that is equivalent to the polygon extruded infinitely along it's face normal.
- Pt **calculateNearestPtOnSurface** (const Pt &a_pt_to_project) const

Return closest Pt to a_pt_to_project that lays on the surface.

Private Types

- using **iterator** = IteratorThroughBracketOperator< Derived >
- using **const_iterator** = ConstIteratorThroughBracketOperator< Derived >

Private Member Functions

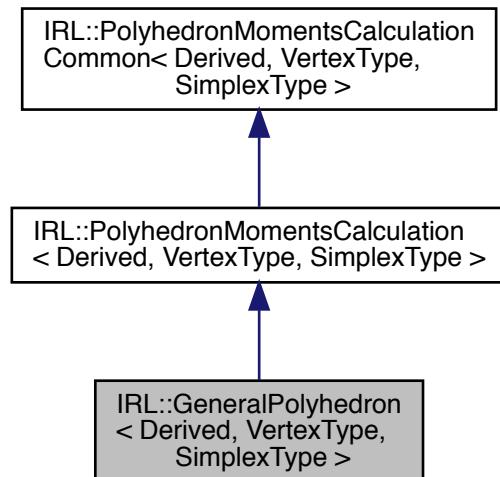
- Derived & **getDerived** (void)
- const Derived & **getDerived** (void) const
- double **calculate2DArea** (const UnsignedIndex_t a_index_0, const UnsignedIndex_t a_index_1) const
Calculate 2D area of polygon projected into 2D plane with greatest magnitude normal component for the plane.
- std::array< UnsignedIndex_t, 3 > **getDimensionsOrderedForAscendingFaceNormalMagnitude** (**Normal** a_normal) const
- **Pt getPtOnPlane** (const **Pt** &a_pt_to_project, const **Normal** &a_polygon_normal) const
- bool **isPtInsidePolygon** (const **Pt** &a_pt, const **Normal** &a_polygon_normal) const
- bool **isPtInternalTo2DPolygon** (const **Pt** &a_pt, const std::array< UnsignedIndex_t, 3 > a_dimension_↔ masking) const
- bool **isPtInBoundingBox** (const **Pt** &a_pt) const
- bool **isPtBeforeIntersectionWithEdge** (const **Pt** &a_test_pt, const **Pt** &a_vertex_0, const **Pt** &a_vertex_1, const std::array< UnsignedIndex_t, 3 > a_dimension_masking) const
- **Pt getNearestProjectedPtOnEdges** (const **Pt** &a_pt_to_project) const
- **Pt getProjectedPtOnEdge** (const **Pt** &a_pt_to_project, const **Pt** &a_edge_pt_0, const **Pt** &a_edge_pt_1) const
- void **removePt** (const UnsignedIndex_t a_index)
- void **takePtIfCloserToOriginalPt** (const **Pt** &a_pt_to_project, **Pt** *a_point_from_the_polygon_edge, **Pt** *a_current_closest_pt, double *a_current_shortest_squared_distance) const

The documentation for this class was generated from the following file:

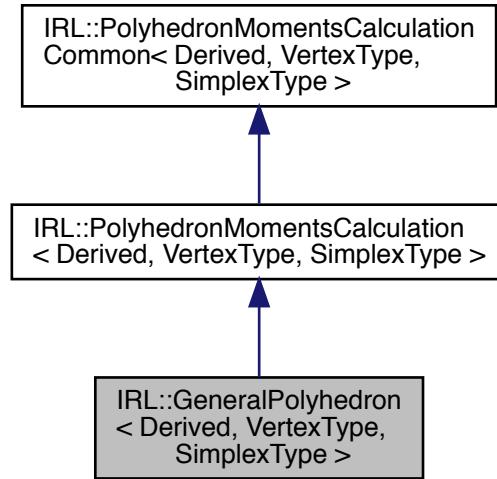
- general_polygon.h

6.443 IRL::GeneralPolyhedron< Derived, VertexType, SimplexType > Class Template Reference

Inheritance diagram for IRL::GeneralPolyhedron< Derived, VertexType, SimplexType >:



Collaboration diagram for IRL::GeneralPolyhedron< Derived, VertexType, SimplexType >:



Public Types

- using **pt_type** = VertexType
- using **value_t** = pt_type

Public Member Functions

- VertexType & **operator[]** (const UnsignedIndex_t a_index)
- const VertexType & **operator[]** (const UnsignedIndex_t a_index) const
- UnsignedIndex_t **getNumberOfSimplicesInDecomposition** (void) const
- SimplexType **getSimplexFromDecomposition** (const UnsignedIndex_t a_tet_number_to_get) const
- UnsignedIndex_t **getNumberOfVertices** (void) const
- **IRL::Pt getLowerLimits** (void) const

Return a point for the lower limits of the polyhedron in 3D space.
- **IRL::Pt getUpperLimits** (void) const

Return a point for the upper limits of the polyhedron in 3D space.
- void **shift** (const double a_x_shift, const double a_y_shift, const double a_z_shift)

Shift entire rectangular cuboid in x, y, and z.
- **iterator begin** (void) noexcept
- **const_iterator begin** (void) const noexcept
- **const_iterator cbegin** (void) const noexcept
- **iterator end** (void) noexcept
- **const_iterator end** (void) const noexcept
- **const_iterator cend** (void) const noexcept

Static Public Member Functions

- template<class OtherPolytope >
static Derived **fromOtherPolytope** (const OtherPolytope &a_other_polytope)
- static constexpr std::array< UnsignedIndex_t, 4 > **getSimplexIndicesFromDecomposition** (const UnsignedIndex_t a_tet)

Private Types

- using **iterator** = [IteratorThroughBracketOperator](#)< Derived >
- using **const_iterator** = [ConstIteratorThroughBracketOperator](#)< Derived >

Private Member Functions

- Derived & **getDerived** (void)
- const Derived & **getDerived** (void) const

The documentation for this class was generated from the following file:

- general_polyhedron.h

6.444 f_cappeddodecahedron_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.445 f_tet_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **tet_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_tet_class.f90](#)

6.446 f_tri_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **tri_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.447 f_dodecahedron_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **dodecahedron_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_dodecahedron_class.f90](#)

6.448 f_cappeddodecahedron_doubles3_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_doubles3_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.449 f_polygon_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **polygon_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.450 f_polyhedron24_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **polyhedron24_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.451 f_polyhedron24_doubles3_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **polyhedron24_doubles3_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.452 f_rectangularcuboid_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **rectangularcuboid_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.453 f_dividedpolygon_class::getboundingpts Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_getboundingpts** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.454 f_vman_class::getcentroid Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **vman_class_getcentroid** (this)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.455 f_sepvm_class::getcentroid Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **sepvm_class_getcentroid** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.456 f_sepvm_doubles3_class::getcentroid Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **sepvm_doubles3_class_getcentroid** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.457 f_tagged_accumvm_sepvm_class::getcentroidatindex Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **tagged_accumvm_sepvm_class_getcentroidatindex** (this, a_list←_index, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.458 `f_tagged_accumvm_vm_class::getcentroidatindex` Interface Reference

Public Member Functions

- `real(irl_double)` function, dimension(3) **tagged_accumvm_vm_class_getcentroidatindex** (`this, a_list_<- index`)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.459 `f_tagged_accumvm_sepvm_class::getcentroidattag` Interface Reference

Public Member Functions

- `real(irl_double)` function, dimension(3) **tagged_accumvm_sepvm_class_getcentroidattag** (`this, a_tag, a_<- _index`)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.460 `f_sepvm_class::getcentroidptr` Interface Reference

Public Member Functions

- `real(irl_double)` function, dimension(:), pointer **sepvm_class_getcentroidptr** (`this, a_index`)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.461 `f_sepvm_doubles3_class::getcentroidptr` Interface Reference

Public Member Functions

- `real(irl_double)` function, dimension(:), pointer **sepvm_doubles3_class_getcentroidptr** (`this, a_index`)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.462 f_tagged_accumvm_vm_class::getcentroidptratindex Interface Reference

Public Member Functions

- real(irl_double) function, dimension(:, pointer **tagged_accumvm_vm_class_getcentroidptratindex** (this, a_list_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.463 f_tagged_accumvm_sepvm_class::getcentroidptratindex Interface Reference

Public Member Functions

- real(irl_double) function, dimension(:, pointer **tagged_accumvm_sepvm_class_getcentroidptratindex** (this, a_list_index, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.464 f_objectallocationserver_localizedseparatorlink_class::getcobject Interface Reference

Public Member Functions

- type([c_objectallocationserver_localizedseparatorlink](#)) function **objectallocationserver_localizedseparatorlink_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_localizedseparatorlink_class.f90](#)

6.465 f_tagged_accumvm_sepvm_class::getcobject Interface Reference

Public Member Functions

- type([c_tagged_accumvm_sepvm](#)) function **tagged_accumvm_sepvm_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.466 f_objectallocationserver_localizerlink_class::getcobject Interface Reference

Public Member Functions

- type([c_objectallocationserver_localizerlink](#)) function **objectallocationserver_localizerlink_class::getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_localizerlink_class.f90](#)

6.467 f_objectallocationserver_planarlocalizer_class::getcobject Interface Reference

Public Member Functions

- type([c_objectallocationserver_planarlocalizer](#)) function **objectallocationserver_planarlocalizer_class::getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_planarlocalizer_class.f90](#)

6.468 f_tagged_accumvm_vm_class::getcobject Interface Reference

Public Member Functions

- type([c_tagged_accumvm_vm](#)) function **tagged_accumvm_vm_class::getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.469 f_objectallocationserver_planarseparator_class::getcobject Interface Reference

Public Member Functions

- type([c_objectallocationserver_planarseparator](#)) function **objectallocationserver_planarseparator_class::getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_planarseparator_class.f90](#)

6.470 f_planarlocalizer_class::getcobject Interface Reference

Public Member Functions

- type([c_planarlocalizer](#)) function **planarlocalizer_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.471 f_tet_class::getcobject Interface Reference

Public Member Functions

- type([c_tet](#)) function **tet_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_tet_class.f90](#)

6.472 f_tri_class::getcobject Interface Reference

Public Member Functions

- type([c_tri](#)) function **tri_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.473 f_planarseparator_class::getcobject Interface Reference

Public Member Functions

- type([c_planarseparator](#)) function **planarseparator_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.474 f_dodecahedron_class::getcobject Interface Reference

Public Member Functions

- type([c_dodecahedron](#)) function **dodecahedron_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_dodecahedron_class.f90](#)

6.475 f_vman_class::getcobject Interface Reference

Public Member Functions

- type([c_vman](#)) function **vman_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.476 f_cappeddodecahedron_doubles3_class::getcobject Interface Reference

Public Member Functions

- type([c_cappeddodecahedron_doubles3](#)) function **cappeddodecahedron_doubles3_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.477 f_elviraneighborhood_class::getcobject Interface Reference

Public Member Functions

- type([c_elviraneighborhood](#)) function **elviraneighborhood_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_elviraneighborhood_class.f90](#)

6.478 f_polygon_class::getcobject Interface Reference

Public Member Functions

- type([c_polygon](#)) function **polygon_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.479 f_polyhedron24_class::getcobject Interface Reference

Public Member Functions

- type([c_polyhedron24](#)) function **polyhedron24_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.480 f_polyhedron24_doubles3_class::getcobject Interface Reference

Public Member Functions

- type([c_polyhedron24_doubles3](#)) function **polyhedron24_doubles3_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.481 f_listedvm_vman_class::getcobject Interface Reference

Public Member Functions

- type([c_listedvm_vman](#)) function **listedvm_vman_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_listedvm_vman_class.f90](#)

6.482 `f_bytebuffer_class::getcobject` Interface Reference

Public Member Functions

- type([c_bytebuffer](#)) function **bytebuffer_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.483 `f_r2pneighborhood_rectangularcuboid_class::getcobject` Interface Reference

Public Member Functions

- type([c_r2pneighborhood_rectangularcuboid](#)) function **r2pneighborhood_rectangularcuboid_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.484 `f_localizedseparatorlink_class::getcobject` Interface Reference

Public Member Functions

- type([c_localizedseparatorlink](#)) function **localizedseparatorlink_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.485 `f_dividedpolygon_class::getcobject` Interface Reference

Public Member Functions

- type([c_dividedpolygon](#)) function **dividedpolygon_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.486 f_rectangularcuboid_class::getcobject Interface Reference

Public Member Functions

- type([c_rectangularcuboid](#)) function **rectangularcuboid_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.487 f_localizerlink_class::getcobject Interface Reference

Public Member Functions

- type([c_localizerlink](#)) function **localizerlink_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.488 f_sepvm_class::getcobject Interface Reference

Public Member Functions

- type([c_sepvm](#)) function **sepvm_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.489 f_sepvm_doubles3_class::getcobject Interface Reference

Public Member Functions

- type([c_sepvm_doubles3](#)) function **sepvm_doubles3_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.490 f_lviraneighborhood_rectangularcuboid_class::getcobject Interface Reference

Public Member Functions

- type([c_lviraneighborhood_rectangularcuboid](#)) function **lviraneighborhood_rectangularcuboid_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_lviraneighborhood_rectangularcuboid_class.f90](#)

6.491 f_tagged_accumlistedvm_vman_class::getcobject Interface Reference

Public Member Functions

- type([c_tagged_accumlistedvm_vman](#)) function **tagged_accumlistedvm_vman_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.492 f_cappeddodecahedron_class::getcobject Interface Reference

Public Member Functions

- type([c_cappeddodecahedron](#)) function **cappeddodecahedron_class_getcobject** (this)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.493 f_cappeddodecahedron_doubles3_class::getdata Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **cappeddodecahedron_doubles3_class_getdata** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.494 f_polyhedron24_doubles3_class::getdata Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **polyhedron24_doubles3_class_getdata** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.495 f_sepvm_doubles3_class::getdata Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **sepvm_doubles3_class_getdata** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.496 f_localizedseparatorlink_class::getid Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **localizedseparatorlink_class_getid** (this)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.497 f_localizerlink_class::getid Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **localizerlink_class_getid** (this)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.498 `f_tagged_accumlistedvm_vman_class::getlistatindex` Interface Reference

Public Member Functions

- subroutine `tagged_accumlistedvm_vman_class_getlistatindex` (this, a_index, a_other_list)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.499 `f_tri_class::getlocalizer` Interface Reference

Public Member Functions

- subroutine `tri_class_getlocalizer` (this, a_planar_localizer)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.500 `f_polygon_class::getlocalizer` Interface Reference

Public Member Functions

- subroutine `polygon_class_getlocalizer` (this, a_planar_localizer)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.501 `f_dividedpolygon_class::getlocalizer` Interface Reference

Public Member Functions

- subroutine `dividedpolygon_class_getlocalizer` (this, a_planar_localizer)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.502 f_listedvm_vman_class::getmoments Interface Reference

Public Member Functions

- subroutine **listedvm_vman_class_getmoments** (this, a_index, a_moments)

The documentation for this interface was generated from the following file:

- [f_listedvm_vman_class.f90](#)

6.503 f_vman_class::getnormal Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **vman_class_getnormal** (this)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.504 f_getvolumemoments::getnormalizedvolumemoments Interface Reference

Public Member Functions

- subroutine **gnvm_d_by_lsl_for_svm** (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **gnvm_cd_by_lsl_for_svm** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **gnvm_cdwd3_by_lsl_for_svmad3** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **gnvm_p24_by_lsl_for_svm** (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)
- subroutine **gnvm_p24wd3_by_lsl_for_svmad3** (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)
- subroutine **gnvm_tet_by_lsl_for_svm** (a_tet, a_localized_separator_link, a_moments_to_return)
- subroutine **gnvm_rc_by_ps_for_v** (a_rectangular_cuboid, a_planar_separator, a_moments_to_return)
- subroutine **gnvm_d_by_ps_for_svm** (a_Dodecahedron, a_planar_separator, a_moments_to_return)
- subroutine **gnvm_cd_by_lsl_for_tagaccumvm_svm** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **gnvm_d_by_lsl_for_tagaccumvm_svm** (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **gnvm_rc_by_ps_for_svm** (a_rectangular_cuboid, a_planar_separator, a_moments_to_return)
- subroutine **gnvm_tri_by_pl_for_v** (a_tri, a_planar_localizer, a_moments_to_return)
- subroutine **gnvm_poly_by_pl_for_v** (a_polygon, a_planar_localizer, a_moments_to_return)
- subroutine **gnvm_tri_by_ll_for_tagavm_vm** (a_tri, a_localizer_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.505 f_planarseparator_class::getnumberofplanes Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **planarseparator_class_getnumberofplanes** (this)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.506 f_polygon_class::getnumberofsimplicesindecomposition Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **polygon_class_getnumberofsimplicesindecomposition** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.507 f_dividedpolygon_class::getnumberofsimplicesindecomposition Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **dividedpolygon_class_getnumberofsimplicesindecomposition** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.508 f_polygon_class::getnumberofvertices Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **polygon_class_getnumberofpts** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.509 f_dividedpolygon_class::getnumberofvertices Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **dividedpolygon_class_getnumberofpts** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.510 f_planarseparator_class::getplane Interface Reference

Public Member Functions

- real(irl_double) function, dimension(4) **planarseparator_class_getplane** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.511 f_tri_class::getplaneofexistence Interface Reference

Public Member Functions

- real(irl_double) function, dimension(4) **tri_class_getplaneofexistence** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.512 f_polygon_class::getplaneofexistence Interface Reference

Public Member Functions

- real(irl_double) function, dimension(4) **polygon_class_getplaneofexistence** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.513 f_dividedpolygon_class::getplaneofexistence Interface Reference

Public Member Functions

- real(irl_double) function, dimension(4) **dividedpolygon_class_getplaneofexistence** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.514 f_cutpolygon::getplanepolygonfromreconstruction Interface Reference

Public Member Functions

- subroutine **getplanepolygonfromreconstruction_rc_poly** (a_rectangular_cuboid, a_planar_separator, a_plane_index, a_polygon)
- subroutine **getplanepolygonfromreconstruction_rc_divpoly** (a_rectangular_cuboid, a_planar_separator, a_plane_index, a_divided_polygon)

The documentation for this interface was generated from the following file:

- [f_cutpolygon.f90](#)

6.515 f_cappeddodecahedron_class::getpt Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **cappeddodecahedron_class_getpt** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_class.f90](#)

6.516 f_cappeddodecahedron_doubles3_class::getpt Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **cappeddodecahedron_doubles3_class_getpt** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.517 f_polygon_class::getpt Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **polygon_class_getpt** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.518 f_polyhedron24_class::getpt Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **polyhedron24_class_getpt** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.519 f_polyhedron24_doubles3_class::getpt Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **polyhedron24_doubles3_class_getpt** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.520 f_dividedpolygon_class::getpt Interface Reference

Public Member Functions

- real(irl_double) function, dimension(3) **dividedpolygon_class_getpt** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.521 f_cutpolygon::getreconstructionsurfacearea Interface Reference

Public Member Functions

- real(irl_double) function **getreconstructionsurfacearea_rc** (a_rectangular_cuboid, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_cutpolygon.f90](#)

6.522 f_polygon_class::getsimplexfromdecomposition Interface Reference

Public Member Functions

- subroutine **polygon_class_getsimplexfromdecomposition** (this, a_tri_number_to_get, a_tri_in←_decomposition)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.523 f_dividedpolygon_class::getsimplexfromdecomposition Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_getsimplexfromdecomposition** (this, a_tri_number_to_get, a_tri_in←_decomposition)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.524 f_tagged_accumvm_vm_class::getsize Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **tagged_accumvm_vm_class_getsize** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.525 f_tagged_accumvm_sepvm_class::getsize Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **tagged_accumvm_sepvm_class_getsize** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.526 f_listedvm_vman_class::getsize Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **listedvm_vman_class_getsize** (this)

The documentation for this interface was generated from the following file:

- [f_listedvm_vman_class.f90](#)

6.527 f_bytebuffer_class::getsize Interface Reference

Public Member Functions

- integer(irl_largeoffsetindex_t) function **bytebuffer_class_getsize** (this)

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.528 f_tagged_accumlistedvm_vman_class::getsize Interface Reference

Public Member Functions

- integer(irl_unsignedindex_t) function **tagged_accumlistedvm_vman_class_getsize** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.529 `f_tagged_accumvm_sepvm_class::gettagforindex` Interface Reference

Public Member Functions

- `integer(irl_unsignedindex_t)` function **`tagged_accumvm_sepvm_class_gettagforindex`** (`this, a_index`)

The documentation for this interface was generated from the following file:

- [`f_tagged_accumvm_sepvm_class.f90`](#)

6.530 `f_tagged_accumvm_vm_class::gettagforindex` Interface Reference

Public Member Functions

- `integer(irl_unsignedindex_t)` function **`tagged_accumvm_vm_class_gettagforindex`** (`this, a_index`)

The documentation for this interface was generated from the following file:

- [`f_tagged_accumvm_vm_class.f90`](#)

6.531 `f_tagged_accumlistedvm_vman_class::gettagforindex` Interface Reference

Public Member Functions

- `integer(irl_unsignedindex_t)` function **`tagged_accumlistedvm_vman_class_gettagforindex`** (`this, a_index`)

The documentation for this interface was generated from the following file:

- [`f_tagged_accumlistedvm_vman_class.f90`](#)

6.532 `f_tri_class::getvertices` Interface Reference

Public Member Functions

- `real(irl_double)` function, dimension(1:3, 1:3) **`tri_class_getvertices`** (`this`)

The documentation for this interface was generated from the following file:

- [`f_tri_class.f90`](#)

6.533 f_vman_class::getvolume Interface Reference

Public Member Functions

- real(irl_double) function **vman_class_getvolume** (this)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.534 f_sepvm_class::getvolume Interface Reference

Public Member Functions

- real(irl_double) function **sepvm_class_getvolume** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.535 f_sepvm_doubles3_class::getvolume Interface Reference

Public Member Functions

- real(irl_double) function **sepvm_doubles3_class_getvolume** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.536 f_tagged_accumvm_sepvm_class::getvolumeatindex Interface Reference

Public Member Functions

- real(irl_double) function **tagged_accumvm_sepvm_class_getvolumeatindex** (this, a_list_index, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.537 f_tagged_accumvm_vm_class::getvolumeatindex Interface Reference

Public Member Functions

- real(irl_double) function **tagged_accumvm_vm_class_getvolumeatindex** (this, a_list_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.538 f_tagged_accumvm_sepvm_class::getvolumeattag Interface Reference

Public Member Functions

- real(irl_double) function **tagged_accumvm_sepvm_class_getvolumeattag** (this, a_tag, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.539 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, Enable > Struct Template Reference

Given a polyhedron and a localizer or separator return the un-normalized ReturnType. (ReturnType can be volume, [VolumeMoments](#), or [SeparatedMoments<VolumeMoments>](#)).

```
#include <generic_cutting.h>
```

6.539.1 Detailed Description

```
template<class ReturnType, class CuttingMethod, class EncompassingType, class ReconstructionType, class Enable = void>
struct IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType,
Enable >
```

Given a polyhedron and a localizer or separator return the un-normalized ReturnType. (ReturnType can be volume, [VolumeMoments](#), or [SeparatedMoments<VolumeMoments>](#)).

The documentation for this struct was generated from the following file:

- [generic_cutting.h](#)

6.540 f_getvolumemoments::getvolumemoments Interface Reference

Public Member Functions

- subroutine **gvm_cd_by_lsl_for_svm** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **gvm_d_by_lsl_for_svm** (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **gvm_p24_by_lsl_for_svm** (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)
- subroutine **gvm_tri_by_ll_for_tagalvm_vman** (a_tri, a_localizer_link, a_moments_to_return)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.541 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, NullReconstruction, enable_if_t< IsNullReconstruction< NullReconstruction >::value > > Struct Template Reference

Static Public Member Functions

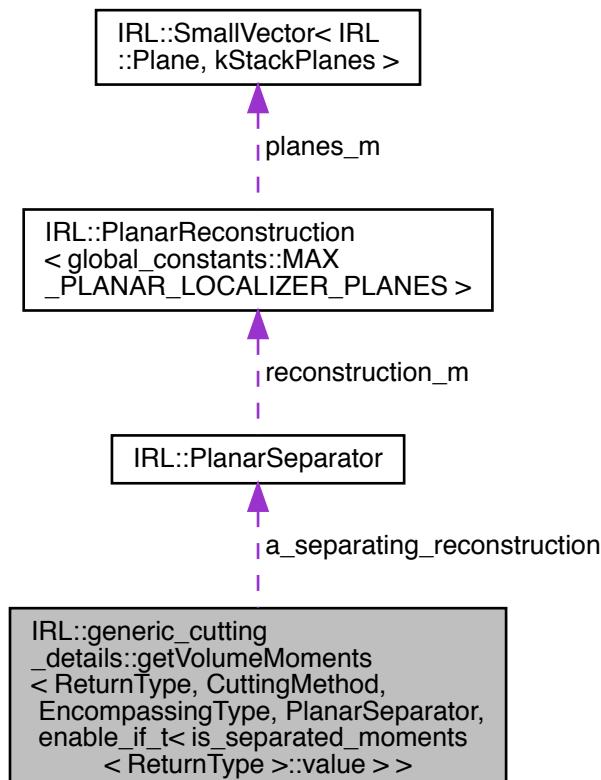
- static ReturnType **getVolumeMomentsImplementation** (const EncompassingType &a_polytope, const NullReconstruction &a_reconstruction)

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.542 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod,
 EncompassingType, PlanarSeparator, enable_if_t< is_separated_moments< ReturnType >::value >>
 Struct Template Reference 309
 6.542 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod,
 EncompassingType, PlanarSeparator, enable_if_t< is_separated_moments< ReturnType >::value >> Struct Template Reference

Collaboration diagram for IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, PlanarSeparator, enable_if_t< is_separated_moments< ReturnType >::value >>:



Public Member Functions

- **__attribute__((pure)) __attribute__((hot))** inline static ReturnType getVolumeMomentsImplementation(const EncompassingType &a_encompassing_polyhedron

Public Attributes

- const PlanarSeparator & **a_separating_reconstruction**

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.543 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isHalfEdgeCutting< CuttingMethod >::value &&!IsNotNullReconstruction< ReconstructionType >::value &&!IsPlanarSeparator< ReconstructionType >::value &&is_separated< moments< ReturnType >::value>> [Struct Template Reference](#)

Public Member Functions

- **__attribute__((pure)) __attribute__((hot)) inline static ReturnType getVolumeMomentsImplementation(const EncompassingType &a_encompassing_polyhedron**

Public Attributes

- **const ReconstructionType & a_separating_reconstruction**

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.544 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isRecursiveSimplex< Cutting< CuttingMethod >::value &&!IsNotNullReconstruction< ReconstructionType >::value &&!IsPlanarSeparator< ReconstructionType >::value &&is_separated< moments< ReturnType >::value>> [Struct Template Reference](#)

Public Member Functions

- **__attribute__((pure)) __attribute__((hot)) inline static ReturnType getVolumeMomentsImplementation(const EncompassingType &a_encompassing_polyhedron**

Public Attributes

- **const ReconstructionType & a_separating_reconstruction**

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.545 IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod,
EncompassingType, ReconstructionType, enable_if_t< isSimplexCutting< CuttingMethod >::value
&&!IsNotNullReconstruction< ReconstructionType >::value &&!(IsPlanarSeparator<
ReconstructionType >::value &&is_separated_moments< ReturnType >::value) >> Struct Template
Reference
**EncompassingType, ReconstructionType, enable_if_t< isSimplexCutting<
CuttingMethod >::value &&!IsNotNullReconstruction< ReconstructionType >::value &&!(IsPlanarSeparator<
ReconstructionType >::value &&is_separated_moments< ReturnType >::value) >> Struct Template Reference³¹¹**

Public Member Functions

- **__attribute__**((pure)) **__attribute__**((hot)) inline static ReturnType getVolumeMomentsImplementation(const
EncompassingType &a_encompassing_polyhedron

Public Attributes

- const ReconstructionType & **a_separating_reconstruction**

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.546 f_getvolumemoments::getvolumemoments_setmethod Interface Reference

Public Member Functions

- subroutine **gvm_setmethod** (a_cutting_method)

The documentation for this interface was generated from the following file:

- [f_getvolumemoments.f90](#)

6.547 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, Reconstruction Type, Enable > Struct Template Reference

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.548 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, NullReconstruction, enable_if_t< IsNullReconstruction< NullReconstruction >>::value >> Struct Template Reference

Static Public Member Functions

- static ReturnType **getVolumeMomentsImplementation** (SegmentedPolytopeType *a_polytope, HalfEdgePolytopeType *a_complete_polytope, const NullReconstruction &a_reconstruction)

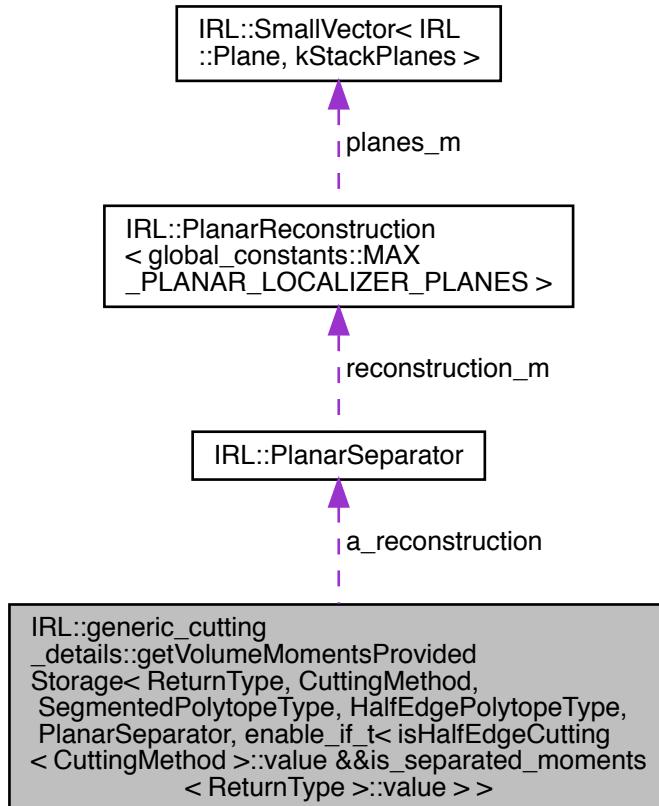
The documentation for this struct was generated from the following file:

- generic_cutting.h

6.549 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, PlanarSeparator, enable_if_t< isHalfEdgeCutting< CuttingMethod >>::value &&is_separated_moments< ReturnType >>::value >> Struct Template Reference

Collaboration diagram for IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, PlanarSeparator, enable_if_t< isHalfEdgeCutting< CuttingMethod >>::value &&is_separated_moments< ReturnType >>::value >>

6.549 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod,
SegmentedPolytopeType, HalfEdgePolytopeType, PlanarSeparator, enable_if_t< isHalfEdgeCutting<
CuttingMethod >::value &&is_separated_moments< ReturnType >::value > > Struct Template Referer 318
Cutting< CuttingMethod >::value &&is_separated_moments< ReturnType >::value >>:



Public Member Functions

- **__attribute__ ((hot))** inline static ReturnType getVolumeMomentsImplementation(SegmentedPolytopeType *a_polytope)

Public Attributes

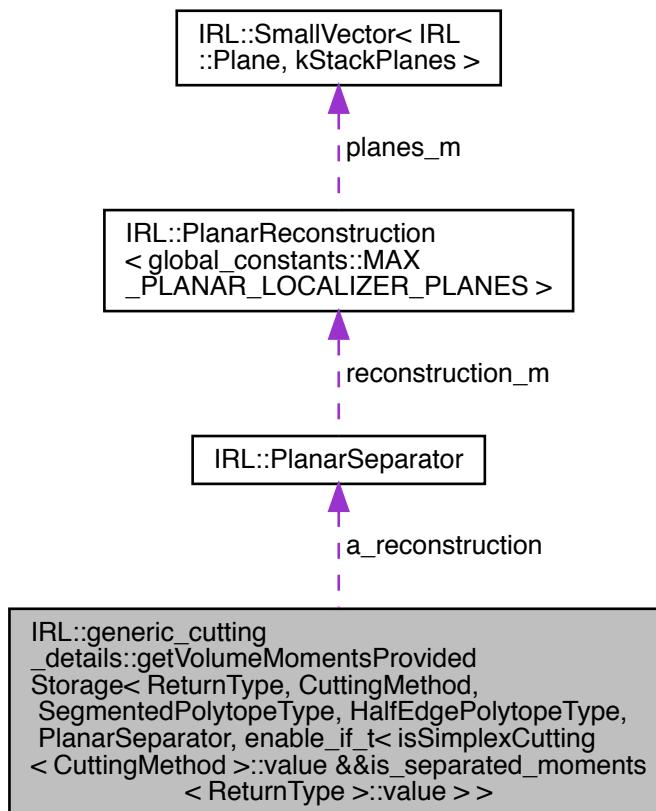
- HalfEdgePolytopeType * **a_complete_polytope**
- HalfEdgePolytopeType const **PlanarSeparator** & **a_reconstruction**

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.550 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, PlanarSeparator, enable_if_t< isSimplexCutting< CuttingMethod >::value &&is_separated_moments< ReturnType >::value > > Struct Template Reference

Collaboration diagram for IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, PlanarSeparator, enable_if_t< isSimplexCutting< CuttingMethod >::value &&is_separated_moments< ReturnType >::value > >:



Public Member Functions

- **__attribute__ ((hot)) inline static ReturnType getVolumeMomentsImplementation(SegmentedPolytopeType *a_polytope)**

Public Attributes

- HalfEdgePolytopeType * **a_complete_polytope**
- HalfEdgePolytopeType const **PlanarSeparator & a_reconstruction**

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.551 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, ReconstructionType, enable_if_t< isHalfEdgeCutting< CuttingMethod >::value && !IsANullReconstruction< ReconstructionType >::value && !IsPlanarSeparator< ReconstructionType >::value && is_separated_moments< ReturnType >::value> > Struct Template Reference³¹⁵

Type, enable_if_t< isHalfEdgeCutting< CuttingMethod >::value && !IsANullReconstruction< ReconstructionType >::value && !IsPlanarSeparator< ReconstructionType >::value && is_separated_moments< ReturnType >::value> > Struct Template Reference

Public Member Functions

- **__attribute__ ((hot)) inline static ReturnType getVolumeMomentsImplementation(SegmentedPolytopeType *a_polytope**

Public Attributes

- HalfEdgePolytopeType * **a_complete_polytope**
- HalfEdgePolytopeType const ReconstructionType & **a_reconstruction**

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.552 IRL::generic_cutting_details::getVolumeMomentsProvidedStorage< ReturnType, CuttingMethod, SegmentedPolytopeType, HalfEdgePolytopeType, ReconstructionType, enable_if_t< isSimplexCutting< CuttingMethod >::value && !IsANullReconstruction< ReconstructionType >::value && !IsPlanarSeparator< ReconstructionType >::value && is_separated_moments< ReturnType >::value> > Struct Template Reference

Public Member Functions

- **__attribute__ ((hot)) inline static ReturnType getVolumeMomentsImplementation(SegmentedPolytopeType *a_polytope**

Public Attributes

- HalfEdgePolytopeType * **a_complete_polytope**
- HalfEdgePolytopeType const ReconstructionType & **a_reconstruction**

The documentation for this struct was generated from the following file:

- generic_cutting.h

6.553 f_sepvm_class::getvolumeptr Interface Reference

Public Member Functions

- real(irl_double) function, pointer **sepvm_class_getvolumeptr** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.554 f_sepvm_doubles3_class::getvolumeptr Interface Reference

Public Member Functions

- real(irl_double) function, pointer **sepvm_doubles3_class_getvolumeptr** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.555 f_tagged_accumvm_vm_class::getvolumepratindex Interface Reference

Public Member Functions

- real(irl_double) function, pointer **tagged_accumvm_vm_class_getvolumepratindex** (this, a_list_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.556 f_tagged_accumvm_sepvm_class::getvolumepratindex Interface Reference

Public Member Functions

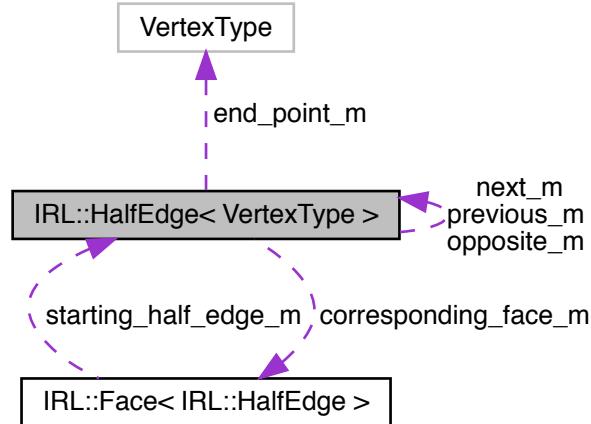
- real(irl_double) function, pointer **tagged_accumvm_sepvm_class_getvolumepratindex** (this, a_list_index, a_index)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.557 IRL::HalfEdge< VertexType > Class Template Reference

Collaboration diagram for IRL::HalfEdge< VertexType >:



Public Types

- using **vertex_type** = VertexType

Public Member Functions

- **HalfEdge** (VertexType *a_vertex, HalfEdge *a_previous, HalfEdge *a_next, Face< HalfEdge > *a_face)
- **HalfEdge** (const HalfEdge &a_other)=default
- **HalfEdge & operator=** (const HalfEdge &a_other)=default
- void **setPreviousHalfEdge** (HalfEdge *a_previous)
- HalfEdge * **getPreviousHalfEdge** (void)
- const HalfEdge * **getPreviousHalfEdge** (void) const
- void **setNextHalfEdge** (HalfEdge *a_next)
- HalfEdge * **getNextHalfEdge** (void)
- const HalfEdge * **getNextHalfEdge** (void) const
- void **setOppositeHalfEdge** (HalfEdge *a_opposite)
- HalfEdge * **getOppositeHalfEdge** (void)
- const HalfEdge * **getOppositeHalfEdge** (void) const
- void **setFace** (Face< HalfEdge > *a_face)
- Face< HalfEdge > * **getFace** (void)
- const Face< HalfEdge > * **getFace** (void) const
- void **setVertex** (VertexType *a_vertex)
- VertexType * **getVertex** (void)
- const VertexType * **getVertex** (void) const
- VertexType * **getPreviousVertex** (void)
- const VertexType * **getPreviousVertex** (void) const

Private Attributes

- `HalfEdge * previous_m`
- `HalfEdge * next_m`
- `HalfEdge * opposite_m`
- `VertexType * end_point_m`
- `Face< HalfEdge > * corresponding_face_m`

The documentation for this class was generated from the following file:

- `half_edge.h`

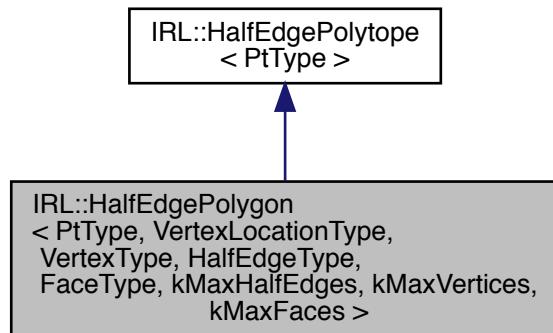
6.558 IRL::HalfEdgeCutting Struct Reference

The documentation for this struct was generated from the following file:

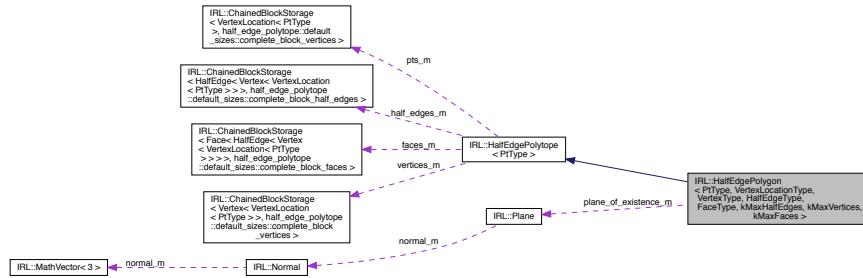
- `default_cutting_method.h`

6.559 IRL::HalfEdgePolygon< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces > Class Template Reference

Inheritance diagram for IRL::HalfEdgePolygon< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >:



Collaboration diagram for IRL::HalfEdgePolygon< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >:



Public Member Functions

- [SegmentedHalfEdgePolygon< FaceType, VertexType > generateSegmentedPolygon \(void\)](#)
- template<class SegmentedType >
[void setSegmentedPolygon \(SegmentedType *a_polytope\)](#)
- [void setPlaneOfExistence \(const \[Plane\]\(#\) &a_plane\)](#)
- [const \[Plane\]\(#\) & getPlaneOfExistence \(void\)](#)

Private Attributes

- [Plane plane_of_existence_m](#)

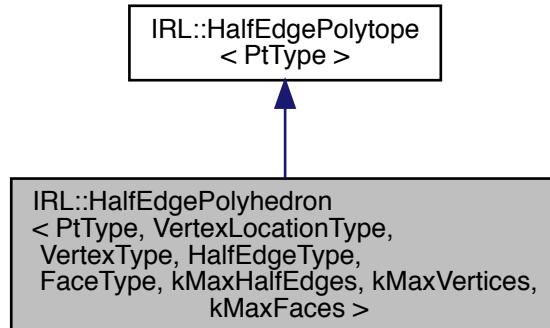
Additional Inherited Members

The documentation for this class was generated from the following file:

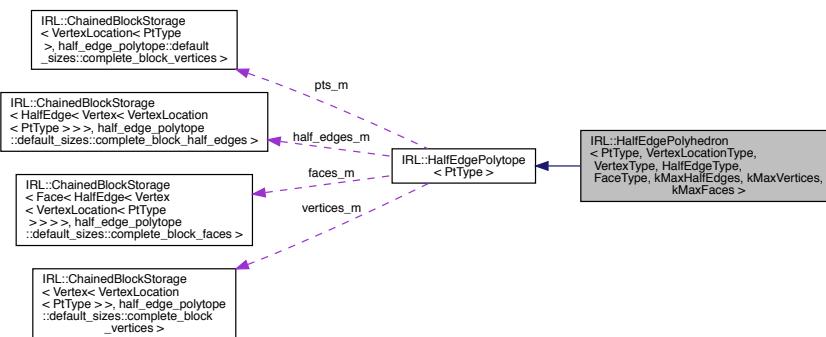
- [half_edge_polygon.h](#)

6.560 IRL::HalfEdgePolyhedron< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces > Class Template Reference

Inheritance diagram for IRL::HalfEdgePolyhedron< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >:



Collaboration diagram for IRL::HalfEdgePolyhedron< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >:



Public Member Functions

- **SegmentedHalfEdgePolyhedron< FaceType, VertexType > generateSegmentedPolyhedron (void)**
- template<class SegmentedType >
void **setSegmentedPolyhedron** (SegmentedType *a_polytope)

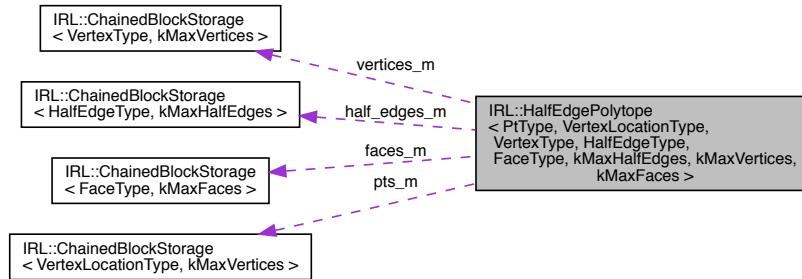
Additional Inherited Members

The documentation for this class was generated from the following file:

- half_edge_polyhedron.h

6.561 IRL::HalfEdgePolytope< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces > Class Template Reference

Collaboration diagram for IRL::HalfEdgePolytope< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >:



Public Types

- using **pt_type** = PtType
- using **vertex_location_type** = VertexLocationType
- using **vertex_type** = VertexType
- using **half_edge_type** = HalfEdgeType
- using **face_type** = FaceType

Public Member Functions

- void **reset** (void)
- void **resize** (const UnsignedIndex_t a_number_of_half_edges, const UnsignedIndex_t a_number_of_vertices, const UnsignedIndex_t a_number_of_faces)
- UnsignedIndex_t **getNumberOfFaces** (void) const
- UnsignedIndex_t **getNumberOfVertices** (void) const
- HalfEdgeType & **getHalfEdge** (const UnsignedIndex_t a_index)
- const HalfEdgeType & **getHalfEdge** (const UnsignedIndex_t a_index) const
- VertexLocationType & **getVertexLocation** (const UnsignedIndex_t a_index)
- const VertexLocationType & **getVertexLocation** (const UnsignedIndex_t a_index) const
- VertexType & **getVertex** (const UnsignedIndex_t a_index)
- const VertexType & **getVertex** (const UnsignedIndex_t a_index) const
- FaceType & **getFace** (const UnsignedIndex_t a_index)
- const FaceType & **getFace** (const UnsignedIndex_t a_index) const
- HalfEdgeType * **getNewHalfEdge** (void)
- HalfEdgeType * **getNewHalfEdge** (HalfEdgeType &&a_half_edge)
- VertexLocationType * **getNewVertexLocation** (void)
- VertexLocationType * **getNewVertexLocation** (VertexLocationType &&a_vertex_location)
- VertexType * **getNewVertex** (void)
- VertexType * **getNewVertex** (VertexType &&a_vertex)
- FaceType * **getNewFace** (void)
- FaceType * **getNewFace** (FaceType &&a_face)
- template<class GeometryType >
- void **setVertexLocations** (const GeometryType &a_geometry)

Static Public Member Functions

- static `HalfEdgePolytope fromKnownSizes` (const `UnsignedIndex_t a_number_of_half_edges`, const `UnsignedIndex_t a_number_of_vertices`, const `UnsignedIndex_t a_number_of_faces`)

Static Public Attributes

- static `constexpr UnsignedIndex_t maxHalfEdges` = `kMaxHalfEdges`
- static `constexpr UnsignedIndex_t maxVertices` = `kMaxVertices`
- static `constexpr UnsignedIndex_t maxFaces` = `kMaxFaces`

Private Member Functions

- `HalfEdgePolytope` (const `UnsignedIndex_t a_number_of_half_edges`, const `UnsignedIndex_t a_number_of_vertices`, const `UnsignedIndex_t a_number_of_faces`)

Private Attributes

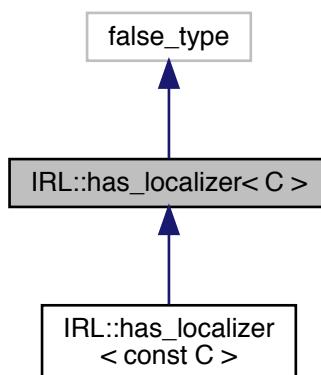
- `ChainedBlockStorage< HalfEdgeType, kMaxHalfEdges > half_edges_m`
- `ChainedBlockStorage< VertexLocationType, kMaxVertices > pts_m`
- `ChainedBlockStorage< VertexType, kMaxVertices > vertices_m`
- `ChainedBlockStorage< FaceType, kMaxFaces > faces_m`

The documentation for this class was generated from the following file:

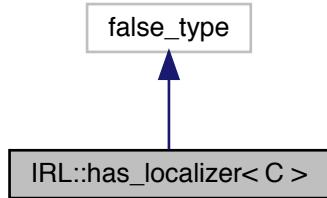
- `half_edge_polytope.h`

6.562 IRL::has_localizer< C > Struct Template Reference

Inheritance diagram for IRL::has_localizer< C >:



Collaboration diagram for IRL::has_localizer< C >:

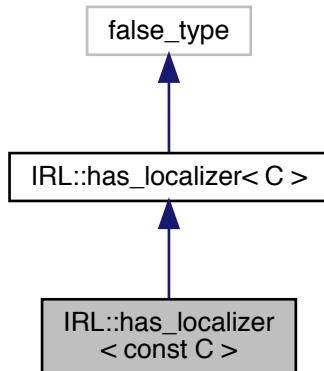


The documentation for this struct was generated from the following file:

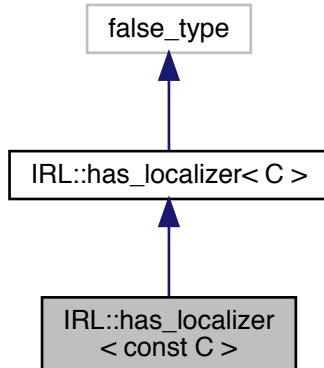
- planar_reconstruction_type_traits.h

6.563 IRL::has_localizer< const C > Struct Template Reference

Inheritance diagram for IRL::has_localizer< const C >:



Collaboration diagram for IRL::has_localizer< const C >:

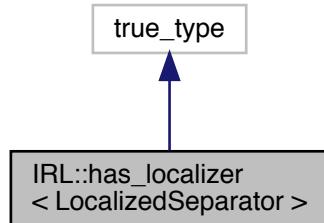


The documentation for this struct was generated from the following file:

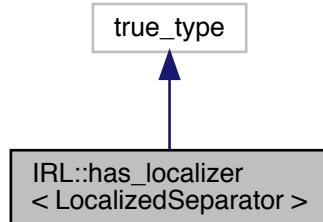
- planar_reconstruction_type_traits.h

6.564 IRL::has_localizer< LocalizedSeparator > Struct Template Reference

Inheritance diagram for IRL::has_localizer< LocalizedSeparator >:



Collaboration diagram for IRL::has_localizer< LocalizedSeparator >:

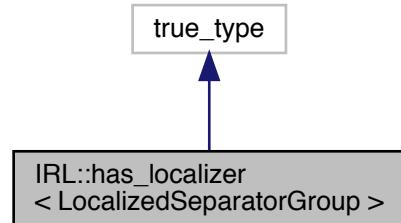


The documentation for this struct was generated from the following file:

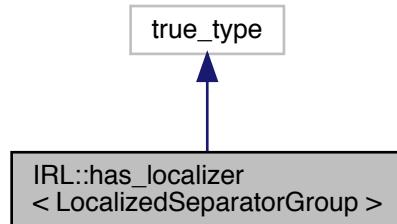
- planar_reconstruction_type_traits.h

6.565 IRL::has_localizer< LocalizedSeparatorGroup > Struct Template Reference

Inheritance diagram for IRL::has_localizer< LocalizedSeparatorGroup >:



Collaboration diagram for IRL::has_localizer< LocalizedSeparatorGroup >:

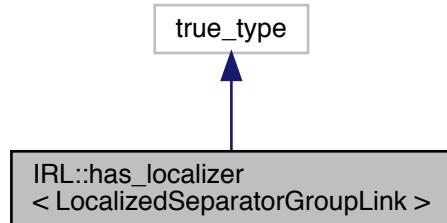


The documentation for this struct was generated from the following file:

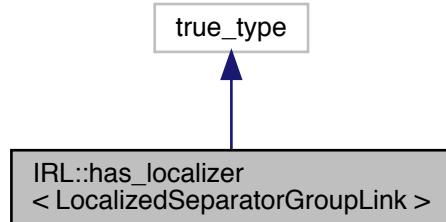
- planar_reconstruction_type_traits.h

6.566 IRL::has_localizer< LocalizedSeparatorGroupLink > Struct Template Reference

Inheritance diagram for IRL::has_localizer< LocalizedSeparatorGroupLink >:



Collaboration diagram for IRL::has_localizer< LocalizedSeparatorGroupLink >:

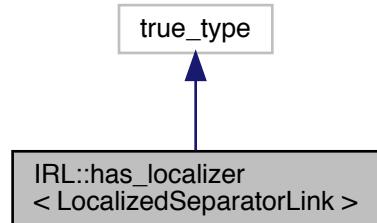


The documentation for this struct was generated from the following file:

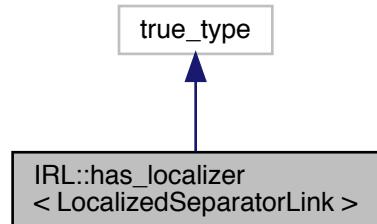
- planar_reconstruction_type_traits.h

6.567 IRL::has_localizer< LocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::has_localizer< LocalizedSeparatorLink >:



Collaboration diagram for IRL::has_localizer< LocalizedSeparatorLink >:

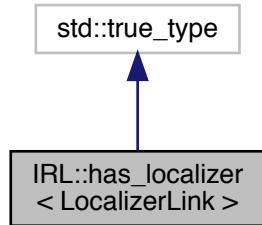


The documentation for this struct was generated from the following file:

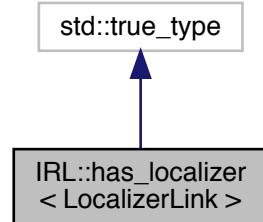
- planar_reconstruction_type_traits.h

6.568 IRL::has_localizer< LocalizerLink > Struct Template Reference

Inheritance diagram for IRL::has_localizer< LocalizerLink >:



Collaboration diagram for IRL::has_localizer< LocalizerLink >:

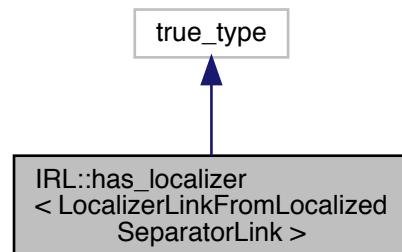


The documentation for this struct was generated from the following file:

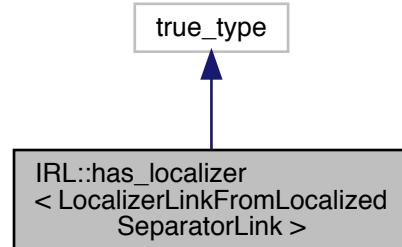
- planar_reconstruction_type_traits.h

6.569 IRL::has_localizer< LocalizerLinkFromLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::has_localizer< LocalizerLinkFromLocalizedSeparatorLink >:



Collaboration diagram for IRL::has_localizer< LocalizerLinkFromLocalizedSeparatorLink >:

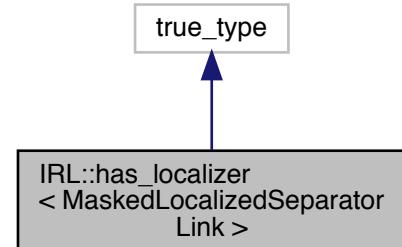


The documentation for this struct was generated from the following file:

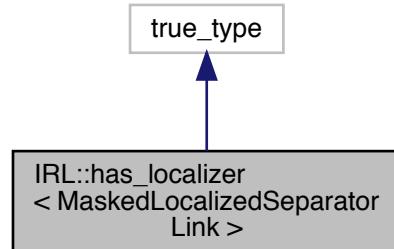
- planar_reconstruction_type_traits.h

6.570 IRL::has_localizer< MaskedLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::has_localizer< MaskedLocalizedSeparatorLink >:



Collaboration diagram for IRL::has_localizer< MaskedLocalizedSeparatorLink >:

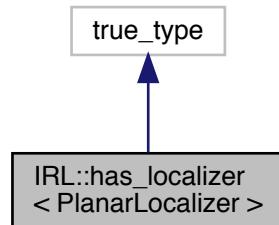


The documentation for this struct was generated from the following file:

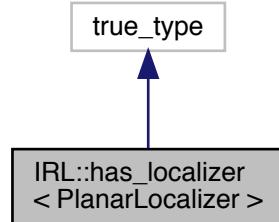
- planar_reconstruction_type_traits.h

6.571 IRL::has_localizer< PlanarLocalizer > Struct Template Reference

Inheritance diagram for IRL::has_localizer< PlanarLocalizer >:



Collaboration diagram for IRL::has_localizer< PlanarLocalizer >:

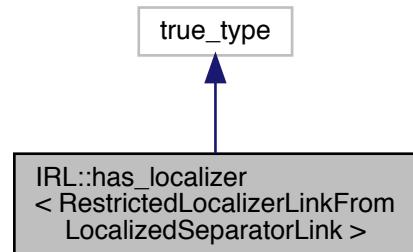


The documentation for this struct was generated from the following file:

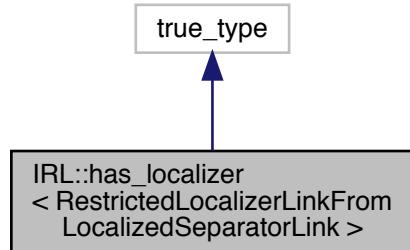
- planar_reconstruction_type_traits.h

6.572 IRL::has_localizer< RestrictedLocalizerLinkFromLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::has_localizer< RestrictedLocalizerLinkFromLocalizedSeparatorLink >:



Collaboration diagram for IRL::has_localizer< RestrictedLocalizerLinkFromLocalizedSeparatorLink >:

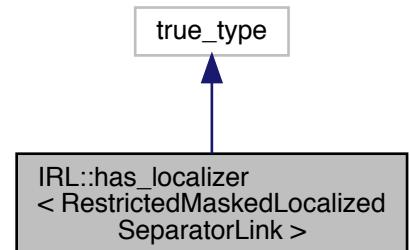


The documentation for this struct was generated from the following file:

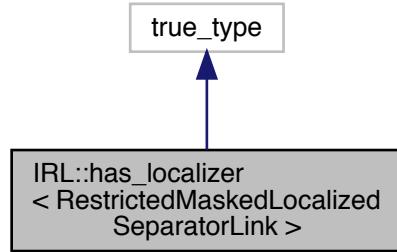
- planar_reconstruction_type_traits.h

6.573 IRL::has_localizer< RestrictedMaskedLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::has_localizer< RestrictedMaskedLocalizedSeparatorLink >:



Collaboration diagram for IRL::has_localizer< RestrictedMaskedLocalizedSeparatorLink >:

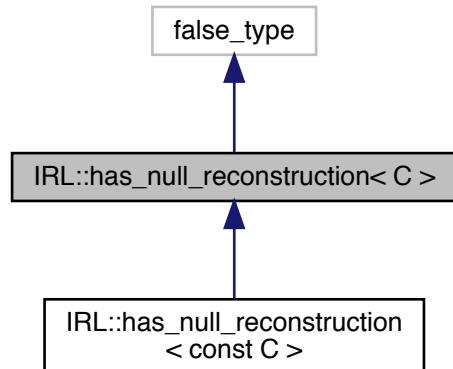


The documentation for this struct was generated from the following file:

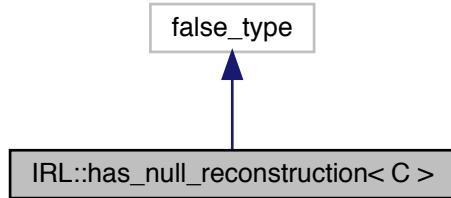
- planar_reconstruction_type_traits.h

6.574 IRL::has_null_reconstruction< C > Struct Template Reference

Inheritance diagram for IRL::has_null_reconstruction< C >:



Collaboration diagram for IRL::has_null_reconstruction< C >:

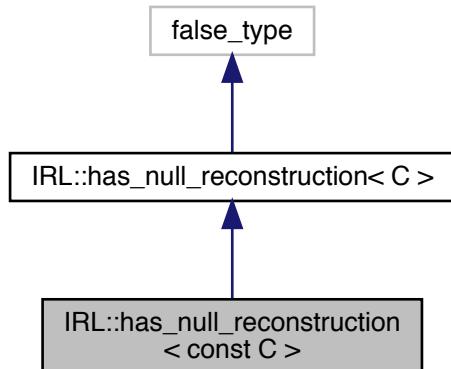


The documentation for this struct was generated from the following file:

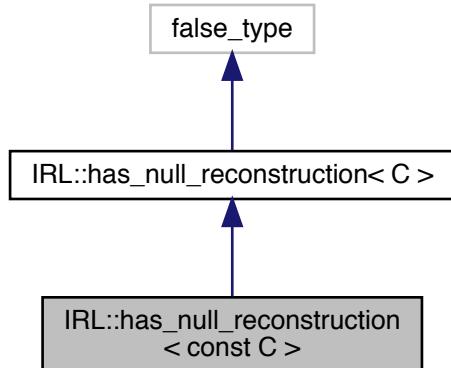
- planar_reconstruction_type_traits.h

6.575 IRL::has_null_reconstruction< const C > Struct Template Reference

Inheritance diagram for IRL::has_null_reconstruction< const C >:



Collaboration diagram for IRL::has_null_reconstruction< const C >:

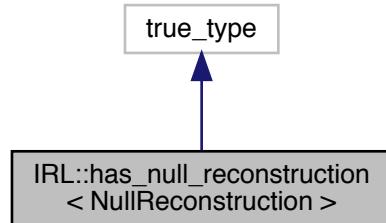


The documentation for this struct was generated from the following file:

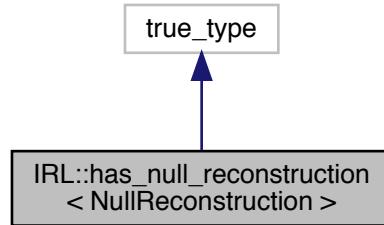
- planar_reconstruction_type_traits.h

6.576 IRL::has_null_reconstruction< NullReconstruction > Struct Template Reference

Inheritance diagram for IRL::has_null_reconstruction< NullReconstruction >:



Collaboration diagram for IRL::has_null_reconstruction< NullReconstruction >:

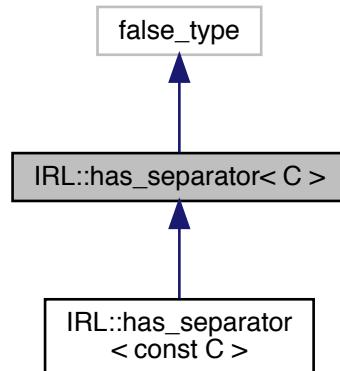


The documentation for this struct was generated from the following file:

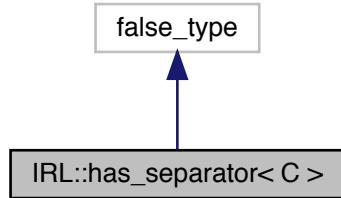
- planar_reconstruction_type_traits.h

6.577 IRL::has_separator< C > Struct Template Reference

Inheritance diagram for IRL::has_separator< C >:



Collaboration diagram for IRL::has_separator< C >:

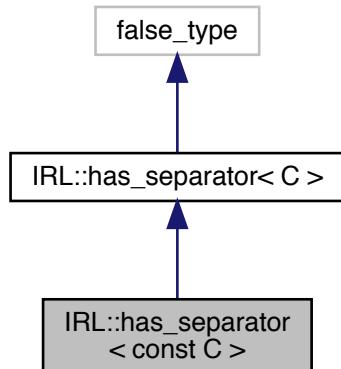


The documentation for this struct was generated from the following file:

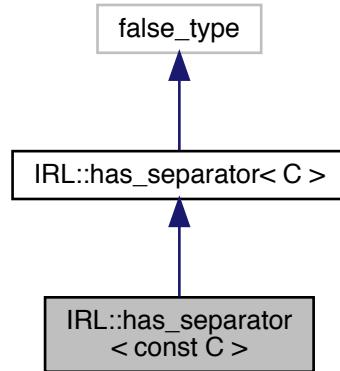
- planar_reconstruction_type_traits.h

6.578 IRL::has_separator< const C > Struct Template Reference

Inheritance diagram for IRL::has_separator< const C >:



Collaboration diagram for IRL::has_separator< const C >:

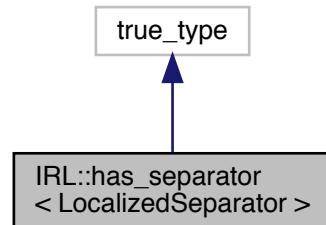


The documentation for this struct was generated from the following file:

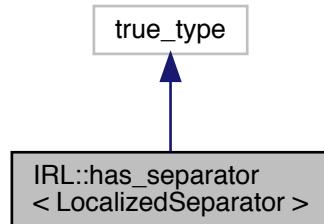
- planar_reconstruction_type_traits.h

6.579 IRL::has_separator< LocalizedSeparator > Struct Template Reference

Inheritance diagram for IRL::has_separator< LocalizedSeparator >:



Collaboration diagram for IRL::has_separator< LocalizedSeparator >:

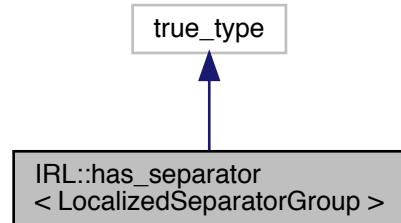


The documentation for this struct was generated from the following file:

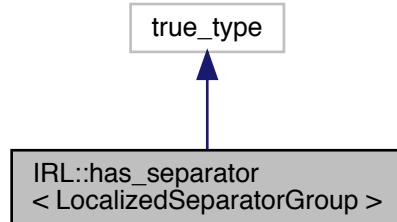
- planar_reconstruction_type_traits.h

6.580 IRL::has_separator< LocalizedSeparatorGroup > Struct Template Reference

Inheritance diagram for IRL::has_separator< LocalizedSeparatorGroup >:



Collaboration diagram for IRL::has_separator< LocalizedSeparatorGroup >:

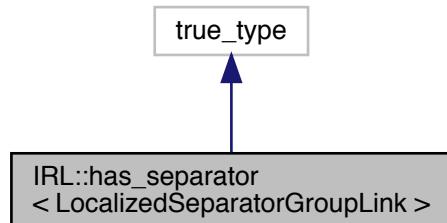


The documentation for this struct was generated from the following file:

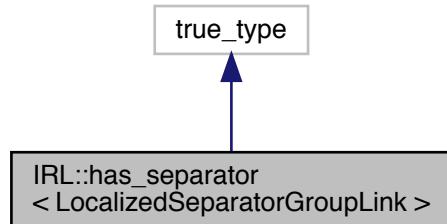
- planar_reconstruction_type_traits.h

6.581 IRL::has_separator< LocalizedSeparatorGroupLink > Struct Template Reference

Inheritance diagram for IRL::has_separator< LocalizedSeparatorGroupLink >:



Collaboration diagram for IRL::has_separator< LocalizedSeparatorGroupLink >:

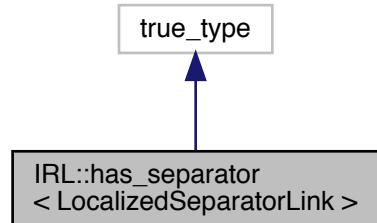


The documentation for this struct was generated from the following file:

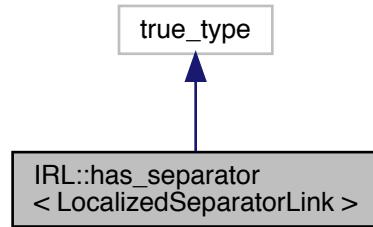
- planar_reconstruction_type_traits.h

6.582 IRL::has_separator< LocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::has_separator< LocalizedSeparatorLink >:



Collaboration diagram for IRL::has_separator< LocalizedSeparatorLink >:

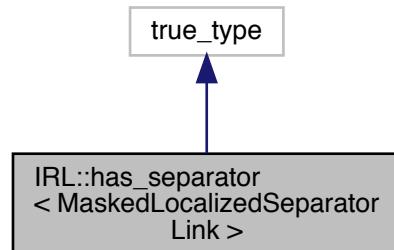


The documentation for this struct was generated from the following file:

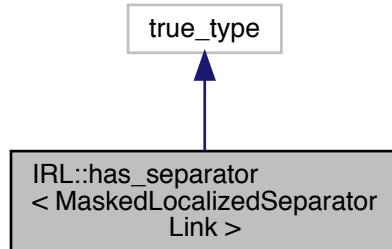
- planar_reconstruction_type_traits.h

6.583 IRL::has_separator< MaskedLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::has_separator< MaskedLocalizedSeparatorLink >:



Collaboration diagram for IRL::has_separator< MaskedLocalizedSeparatorLink >:

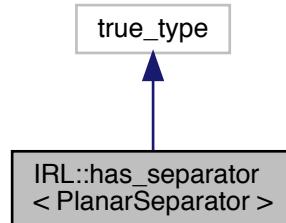


The documentation for this struct was generated from the following file:

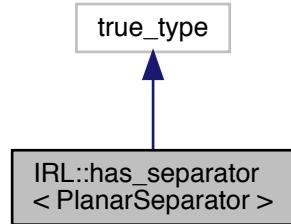
- planar_reconstruction_type_traits.h

6.584 IRL::has_separator< PlanarSeparator > Struct Template Reference

Inheritance diagram for IRL::has_separator< PlanarSeparator >:



Collaboration diagram for IRL::has_separator< PlanarSeparator >:

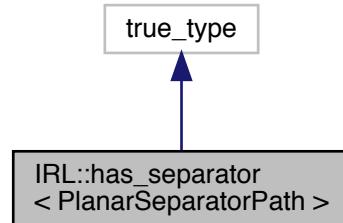


The documentation for this struct was generated from the following file:

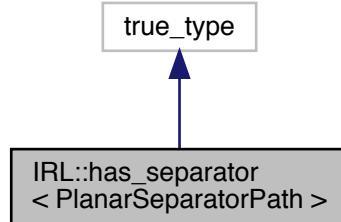
- planar_reconstruction_type_traits.h

6.585 IRL::has_separator< PlanarSeparatorPath > Struct Template Reference

Inheritance diagram for IRL::has_separator< PlanarSeparatorPath >:



Collaboration diagram for IRL::has_separator< PlanarSeparatorPath >:

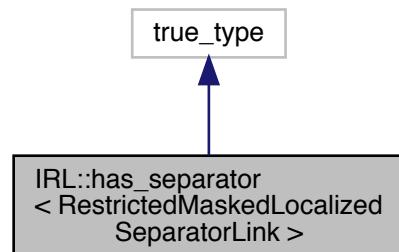


The documentation for this struct was generated from the following file:

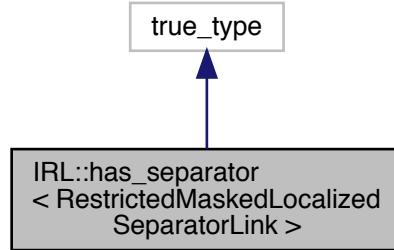
- planar_reconstruction_type_traits.h

6.586 IRL::has_separator< RestrictedMaskedLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::has_separator< RestrictedMaskedLocalizedSeparatorLink >:



Collaboration diagram for IRL::has_separator< RestrictedMaskedLocalizedSeparatorLink >:



The documentation for this struct was generated from the following file:

- planar_reconstruction_type_traits.h

6.587 IRL::HasACollection< ReturnType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value** = [is_moments_collection<ReturnType>::value](#)

The documentation for this struct was generated from the following file:

- class_classifications.h

6.588 IRL::HasALocalizer< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value** = [has_localizer<ReconstructionType>::value](#)

The documentation for this struct was generated from the following file:

- class_classifications.h

6.589 IRL::HasANestedType< ReturnType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value** = [is_nested_moments](#)<ReturnType>::value

The documentation for this struct was generated from the following file:

- class_classifications.h

6.590 IRL::HasAReconstructionLink< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value**

6.590.1 Member Data Documentation

6.590.1.1 value

```
template<class ReconstructionType >
constexpr bool IRL::HasAReconstructionLink< ReconstructionType >::value [static]
```

Initial value:

```
=  
    is_reconstruction_link<ReconstructionType>::value
```

The documentation for this struct was generated from the following file:

- class_classifications.h

6.591 IRL::HasASeparator< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value** = [has_separator](#)<ReconstructionType>::value

The documentation for this struct was generated from the following file:

- class_classifications.h

6.592 IRL::HasLocalizer_AND_HasSeparator< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value**

6.592.1 Member Data Documentation

6.592.1.1 value

```
template<class ReconstructionType >
constexpr bool IRL::HasLocalizer_AND_HasSeparator< ReconstructionType >::value [static]
```

Initial value:

```
= (HasALocalizer<ReconstructionType>::value &&
    HasASeparator<ReconstructionType>::value)
```

The documentation for this struct was generated from the following file:

- class_classifications.h

6.593 IRL::HasLocalizer_OR_HasSeparator< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value**

6.593.1 Member Data Documentation

6.593.1.1 value

```
template<class ReconstructionType >
constexpr bool IRL::HasLocalizer_OR_HasSeparator< ReconstructionType >::value [static]
```

Initial value:

```
= (HasALocalizer<ReconstructionType>::value ||
    HasASeparator<ReconstructionType>::value)
```

The documentation for this struct was generated from the following file:

- class_classifications.h

6.594 IRL::HasLocalizer_XOR_HasSeparator< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value**

6.594.1 Member Data Documentation

6.594.1.1 value

```
template<class ReconstructionType >
constexpr bool IRL::HasLocalizer_XOR_HasSeparator< ReconstructionType >::value [static]
```

Initial value:

```
=
(HasLocalizer_OR_HasSeparator<ReconstructionType>::value &&
!HasLocalizer_AND_HasSeparator<ReconstructionType>::value)
```

The documentation for this struct was generated from the following file:

- class_classifications.h

6.595 IRL::HexahedronSpecialization< Derived, VertexType > Class Template Reference

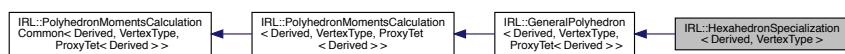
A hexahedron class.

```
#include <hexahedron.h>
```

Inheritance diagram for IRL::HexahedronSpecialization< Derived, VertexType >:



Collaboration diagram for IRL::HexahedronSpecialization< Derived, VertexType >:



Public Member Functions

- `HalfEdgePolyhedron< VertexType > generateHalfEdgeVersion (void) const`
- template<class HalfEdgePolyhedronType >
`void setHalfEdgeVersion (HalfEdgePolyhedronType *a_half_edge_version) const`
- `PlanarLocalizer getLocalizer (void) const`
Returns a planar reconstruction that is equivalent to the HexahedronCommon.
- `ProxyTet< Derived > getSimplexFromDecomposition (const UnsignedIndex_t a_tet) const`

Static Public Member Functions

- static constexpr UnsignedIndex_t `getNumberOfSimplicesInDecomposition (void)`
- static constexpr std::array< UnsignedIndex_t, 4 > `getSimplexIndicesFromDecomposition (const UnsignedIndex_t a_tet)`

Additional Inherited Members

6.595.1 Detailed Description

```
template<class Derived, class VertexType>
class IRL::HexahedronSpecialization< Derived, VertexType >
```

A hexahedron class.

The documentation for this class was generated from the following file:

- `hexahedron.h`

6.596 IRL::IgnoreVolumeAbovePlane Struct Reference

Static Public Member Functions

- template<class SimplexType , class ReconstructionType , class VerticesList , class ReturnType >
`static void generateAndHandleSimplicesFromVolumeAbovePlane (const ReconstructionType &a_reconstruction, const LookupIndex_t a_cutting_case_for_simplex_and_current_plane, const VerticesList &a_simplex_vertices_and_intersection_points, const UnsignedIndex_t a_index_for_plane_that_just_cut, ReturnType *a_moments_to_return)`

The documentation for this struct was generated from the following file:

- `continue_dividing_volume.h`

6.597 IRL::Illinois< OptimizingClass > Class Template Reference

A templated class driver that performs the [Illinois](#) method of Regula-Falsi zero-finding.

```
#include <illinois.h>
```

Public Member Functions

- `Illinois (void)`
Default constructor.
- `void solve (OptimizingClass *a_otype, double a_bracket_0, double a_bracket_1)`
Solution that resets optimizing class and brackets, then solves.
- `~Illinois (void)=default`
Default destructor.

Private Member Functions

- `void solve (double a_bracket_0, double a_bracket_1)`
Given two locations that bracket a zero, find the zero.
- `double calculateError (double *a_guess)`
Set the guess and calculate the associated error.

Private Attributes

- `OptimizingClass * otype_m`
Pointer to object of class OptimizingClass that is being optimized.

6.597.1 Detailed Description

```
template<class OptimizingClass>
class IRL::Illinois< OptimizingClass >
```

A templated class driver that performs the `Illinois` method of Regula-Falsi zero-finding.

This class is a general Regula-Falsi solver that takes a pointer to a templated class that must implement certain methods (detailed below). Upon completion, the final solution should be stored in the pointed-to-class because of the use of `updateGuess ()`.

Requirements for Optimizing class: `calculateSignedScalarError ()` : A method to calculate a scalar error for the class with sign of (correct-guess).

- `updateGuess (....)` : A method that takes in the delta change and computes a new guess vector (which it is storing itself)
- `updateBoundaries (....)` : (Only if `solveAndTrackBounds ()` is being used) A method to keep track of the best current solutions on the +/- side of the zero.
- `errorTooHigh (....)` : A method that takes error and returns a boolean whether the error is low enough to stop optimization and return.

6.597.2 Member Function Documentation

6.597.2.1 `solve()` [1/2]

```
template<class OptimizingClass >
void IRL::Illinois< OptimizingClass >::solve (
    OptimizingClass * a_otype,
    double a_bracket_0,
    double a_bracket_1 )
```

Solution that resets optimizing class and brackets, then solves.

Parameters

in	<i>a_otype</i>	Pointer to class being optimized
in	<i>a_bracket_0</i>	Bracket on one side of zero.
in	<i>a_bracket_1</i>	Bracket on other side of zero from <i>a_bracket_0</i> .

6.597.2.2 solve() [2/2]

```
template<class OptimizingClass >
void IRL::Illinois< OptimizingClass >::solve (
    double a_bracket_0,
    double a_bracket_1 ) [private]
```

Given two locations that bracket a zero, find the zero.

Parameters

in	<i>a_bracket_0</i>	Bracket on one side of zero.
in	<i>a_bracket_1</i>	Bracket on other side of zero from <i>a_bracket_0</i> .

The documentation for this class was generated from the following file:

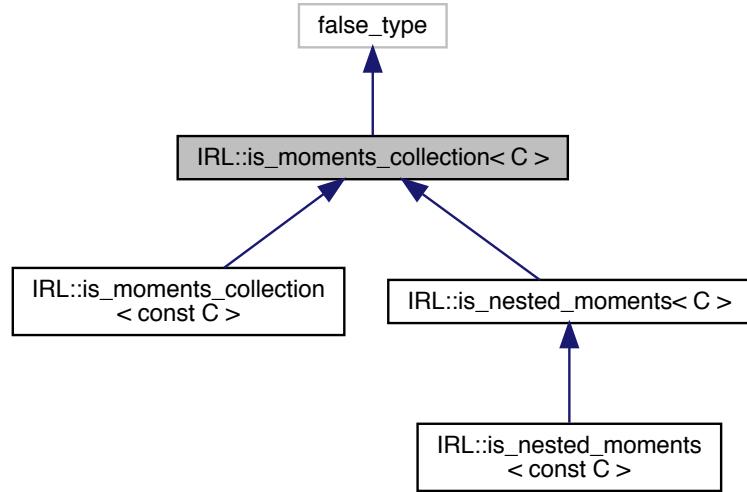
- illinois.h

6.598 IRL::is_moments_collection< C > Struct Template Reference

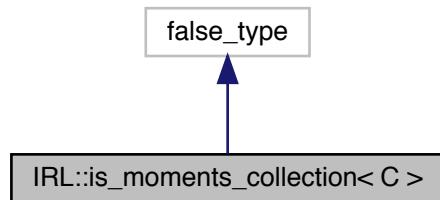
Type trait to allow static checking that an object is a [AccumulatedVolumeMoments](#).

```
#include <moments_type_traits.h>
```

Inheritance diagram for IRL::is_moments_collection< C >:



Collaboration diagram for IRL::is_moments_collection< C >:



6.598.1 Detailed Description

```
template<class C>
struct IRL::is_moments_collection< C >
```

Type trait to allow static checking that an object is a [AccumulatedVolumeMoments](#).

The documentation for this struct was generated from the following file:

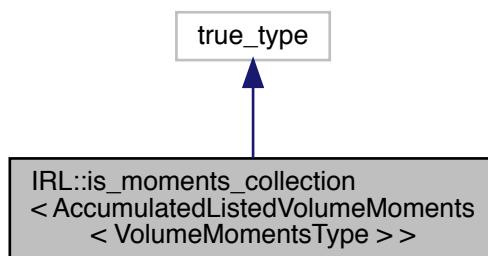
- moments_type_traits.h

6.599 IRL::is_moments_collection< AccumulatedListedVolumeMoments< VolumeMomentsType > > Struct Template Reference

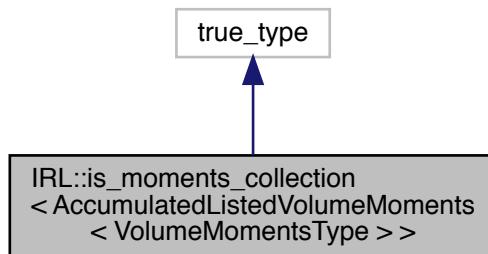
Any instantiation of [AccumulatedListedVolumeMoments](#) is a type of volumeMomentsList.

```
#include <moments_type_traits.h>
```

Inheritance diagram for IRL::is_moments_collection< AccumulatedListedVolumeMoments< VolumeMomentsType > >:



Collaboration diagram for IRL::is_moments_collection< AccumulatedListedVolumeMoments< VolumeMomentsType > >:



6.599.1 Detailed Description

```
template<class VolumeMomentsType>
struct IRL::is_moments_collection< AccumulatedListedVolumeMoments< VolumeMomentsType > >
```

Any instantiation of [AccumulatedListedVolumeMoments](#) is a type of volumeMomentsList.

The documentation for this struct was generated from the following file:

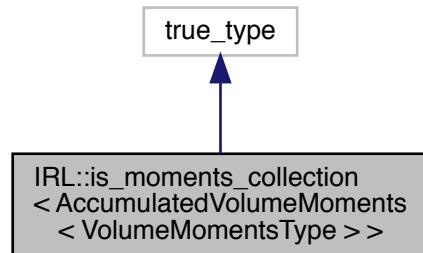
- moments_type_traits.h

6.600 IRL::is_moments_collection< AccumulatedVolumeMoments< VolumeMomentsType > > Struct Template Reference

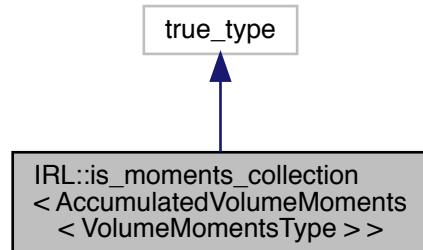
Any instantiation of [AccumulatedVolumeMoments](#) is a type of volumeMomentsList.

```
#include <moments_type_traits.h>
```

Inheritance diagram for IRL::is_moments_collection< AccumulatedVolumeMoments< VolumeMomentsType > >:



Collaboration diagram for IRL::is_moments_collection< AccumulatedVolumeMoments< VolumeMomentsType > >:



6.600.1 Detailed Description

```
template<class VolumeMomentsType>
struct IRL::is_moments_collection< AccumulatedVolumeMoments< VolumeMomentsType > >
```

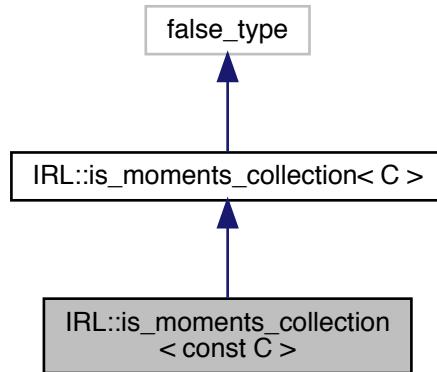
Any instantiation of [AccumulatedVolumeMoments](#) is a type of volumeMomentsList.

The documentation for this struct was generated from the following file:

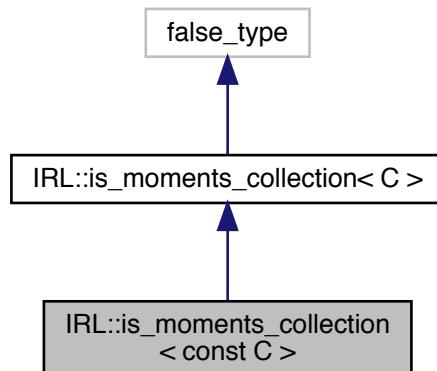
- moments_type_traits.h

6.601 IRL::is_moments_collection< const C > Struct Template Reference

Inheritance diagram for IRL::is_moments_collection< const C >:



Collaboration diagram for IRL::is_moments_collection< const C >:



The documentation for this struct was generated from the following file:

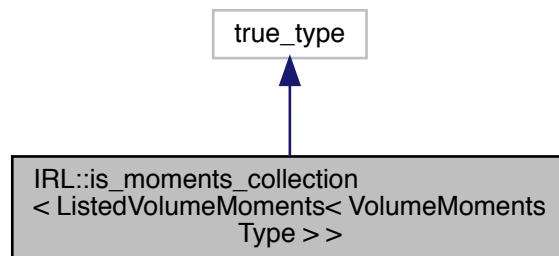
- moments_type_traits.h

6.602 IRL::is_moments_collection< ListedVolumeMoments< VolumeMomentsType > > Struct Template Reference

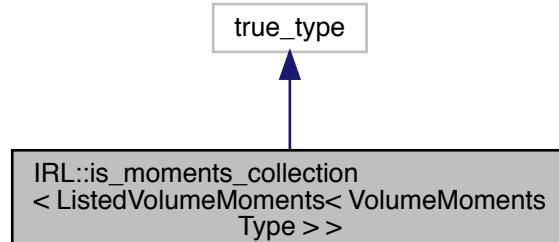
Any instantiation of [ListedVolumeMoments](#) is a type of volumeMomentsList.

```
#include <moments_type_traits.h>
```

Inheritance diagram for IRL::is_moments_collection< ListedVolumeMoments< VolumeMomentsType > >:



Collaboration diagram for IRL::is_moments_collection< ListedVolumeMoments< VolumeMomentsType > >:



6.602.1 Detailed Description

```
template<class VolumeMomentsType>
struct IRL::is_moments_collection< ListedVolumeMoments< VolumeMomentsType > >
```

Any instantiation of [ListedVolumeMoments](#) is a type of volumeMomentsList.

The documentation for this struct was generated from the following file:

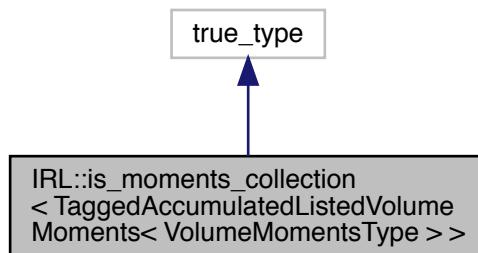
- moments_type_traits.h

6.603 IRL::is_moments_collection< TaggedAccumulatedListedVolumeMoments< VolumeMomentsType > > Struct Template Reference

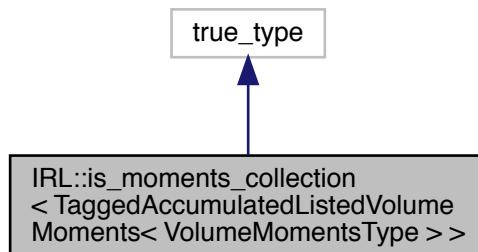
Any instantiation of [AccumulatedListedVolumeMoments](#) is a type of volumeMomentsList.

```
#include <moments_type_traits.h>
```

Inheritance diagram for IRL::is_moments_collection< TaggedAccumulatedListedVolumeMoments< VolumeMomentsType > >:



Collaboration diagram for IRL::is_moments_collection< TaggedAccumulatedListedVolumeMoments< VolumeMomentsType > >:



6.603.1 Detailed Description

```
template<class VolumeMomentsType>
struct IRL::is_moments_collection< TaggedAccumulatedListedVolumeMoments< VolumeMomentsType > >
```

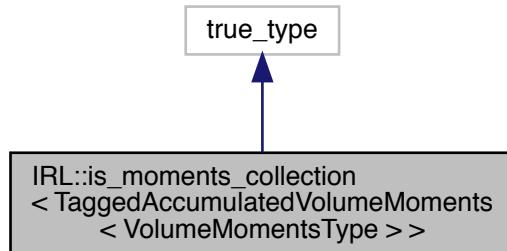
Any instantiation of [AccumulatedListedVolumeMoments](#) is a type of volumeMomentsList.

The documentation for this struct was generated from the following file:

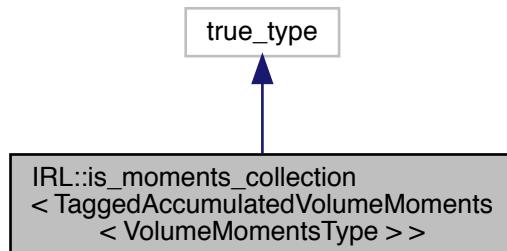
- moments_type_traits.h

6.604 IRL::is_moments_collection< TaggedAccumulatedVolumeMoments< VolumeMomentsType > > Struct Template Reference

Inheritance diagram for IRL::is_moments_collection< TaggedAccumulatedVolumeMoments< VolumeMomentsType > >:



Collaboration diagram for IRL::is_moments_collection< TaggedAccumulatedVolumeMoments< VolumeMomentsType > >:



The documentation for this struct was generated from the following file:

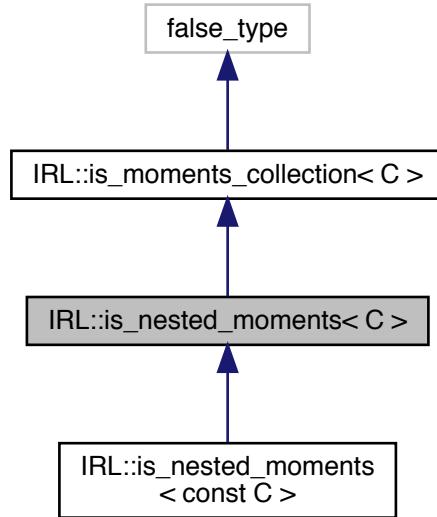
- moments_type_traits.h

6.605 IRL::is_nested_moments< C > Struct Template Reference

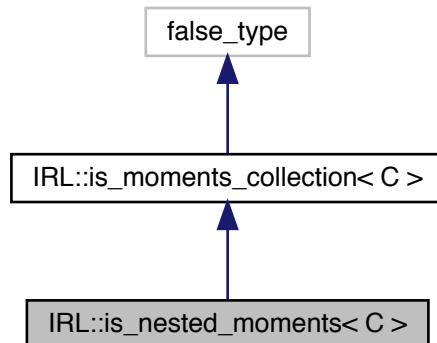
Type trait to allow static checking that an object is a [AccumulatedVolumeMoments](#).

```
#include <moments_type_traits.h>
```

Inheritance diagram for IRL::is_nested_moments< C >:



Collaboration diagram for IRL::is_nested_moments< C >:



6.605.1 Detailed Description

```
template<class C>
struct IRL::is_nested_moments< C >
```

Type trait to allow static checking that an object is a [AccumulatedVolumeMoments](#).

The documentation for this struct was generated from the following file:

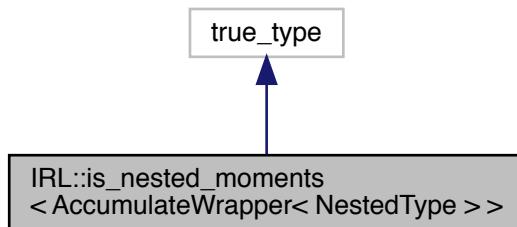
- moments_type_traits.h

6.606 IRL::is_nested_moments< AccumulateWrapper< NestedType > > Struct Template Reference

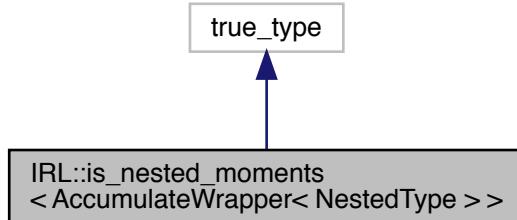
Any instantiation of [AccumulatedVolumeMoments](#) is a type of volumeMomentsList.

```
#include <moments_type_traits.h>
```

Inheritance diagram for IRL::is_nested_moments< AccumulateWrapper< NestedType > >:



Collaboration diagram for IRL::is_nested_moments< AccumulateWrapper< NestedType > >:



6.606.1 Detailed Description

```
template<class NestedType>
struct IRL::is_nested_moments< AccumulateWrapper< NestedType > >
```

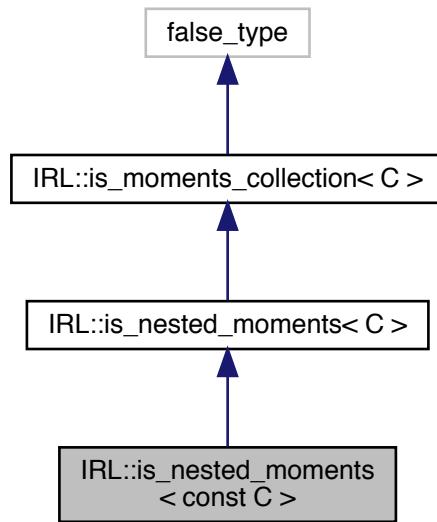
Any instantiation of [AccumulatedVolumeMoments](#) is a type of volumeMomentsList.

The documentation for this struct was generated from the following file:

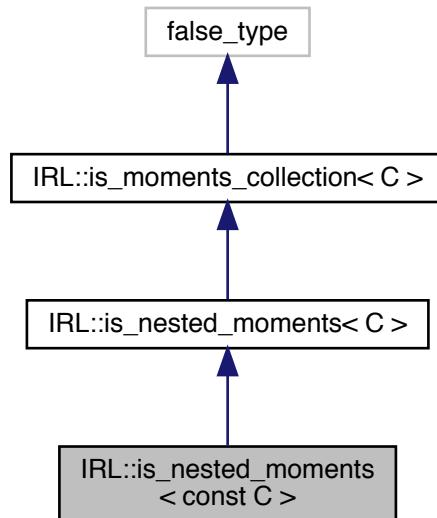
- moments_type_traits.h

6.607 IRL::is_nested_moments< const C > Struct Template Reference

Inheritance diagram for IRL::is_nested_moments< const C >:



Collaboration diagram for IRL::is_nested_moments< const C >:

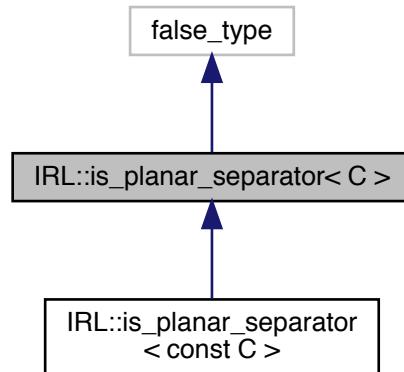


The documentation for this struct was generated from the following file:

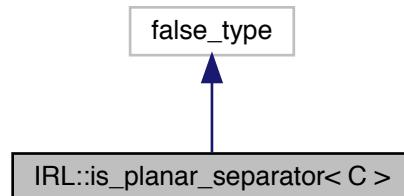
- moments_type_traits.h

6.608 IRL::is_planar_separator< C > Struct Template Reference

Inheritance diagram for IRL::is_planar_separator< C >:



Collaboration diagram for IRL::is_planar_separator< C >:

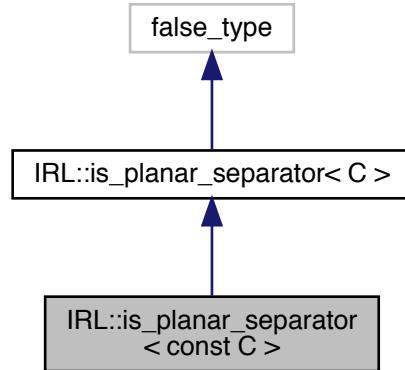


The documentation for this struct was generated from the following file:

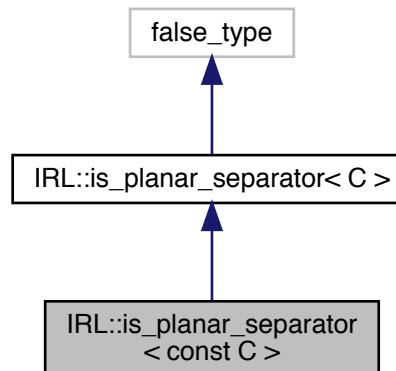
- planar_reconstruction_type_traits.h

6.609 IRL::is_planar_separator< const C > Struct Template Reference

Inheritance diagram for IRL::is_planar_separator< const C >:



Collaboration diagram for IRL::is_planar_separator< const C >:

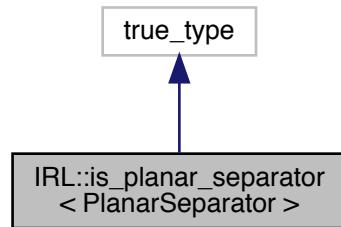


The documentation for this struct was generated from the following file:

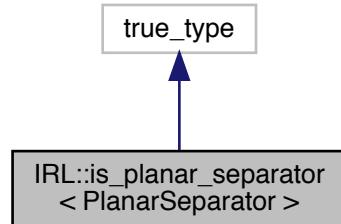
- planar_reconstruction_type_traits.h

6.610 IRL::is_planar_separator< PlanarSeparator > Struct Template Reference

Inheritance diagram for IRL::is_planar_separator< PlanarSeparator >:



Collaboration diagram for IRL::is_planar_separator< PlanarSeparator >:

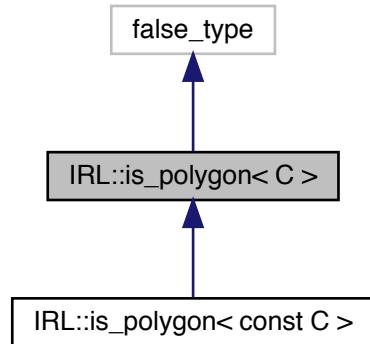


The documentation for this struct was generated from the following file:

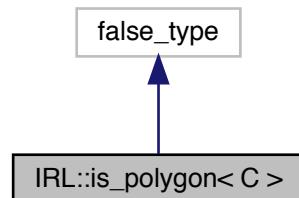
- planar_reconstruction_type_traits.h

6.611 IRL::is_polygon< C > Struct Template Reference

Inheritance diagram for IRL::is_polygon< C >:



Collaboration diagram for IRL::is_polygon< C >:

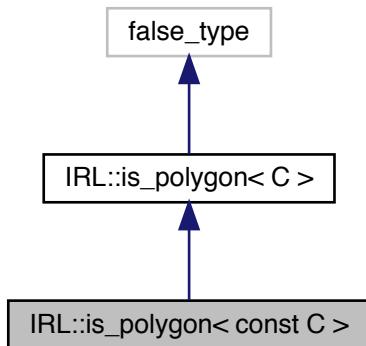


The documentation for this struct was generated from the following file:

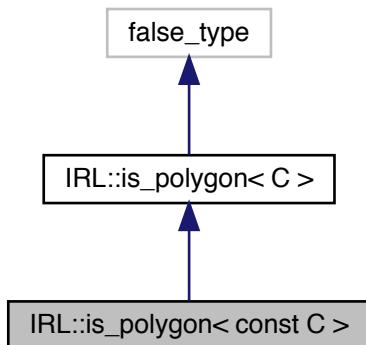
- geometry_type_traits.h

6.612 IRL::is_polygon< const C > Struct Template Reference

Inheritance diagram for IRL::is_polygon< const C >:



Collaboration diagram for IRL::is_polygon< const C >:

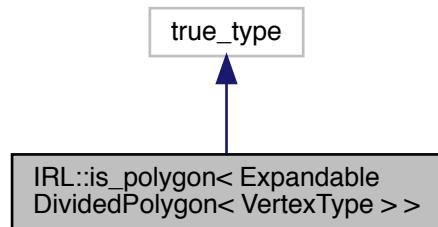


The documentation for this struct was generated from the following file:

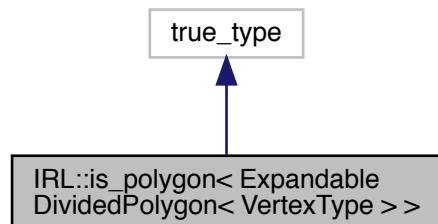
- geometry_type_traits.h

6.613 IRL::is_polygon< ExpandableDividedPolygon< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polygon< ExpandableDividedPolygon< VertexType > >:



Collaboration diagram for IRL::is_polygon< ExpandableDividedPolygon< VertexType > >:

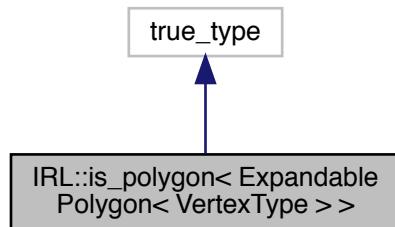


The documentation for this struct was generated from the following file:

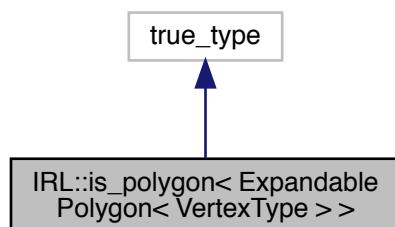
- geometry_type_traits.h

6.614 IRL::is_polygon< ExpandablePolygon< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polygon< ExpandablePolygon< VertexType > >:



Collaboration diagram for IRL::is_polygon< ExpandablePolygon< VertexType > >:

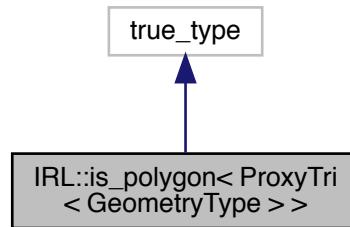


The documentation for this struct was generated from the following file:

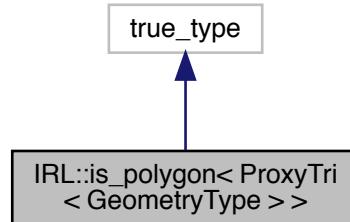
- geometry_type_traits.h

6.615 IRL::is_polygon< ProxyTri< GeometryType > > Struct Template Reference

Inheritance diagram for IRL::is_polygon< ProxyTri< GeometryType > >:



Collaboration diagram for IRL::is_polygon< ProxyTri< GeometryType > >:

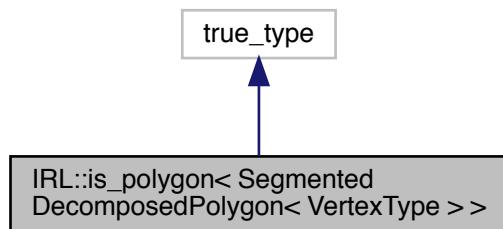


The documentation for this struct was generated from the following file:

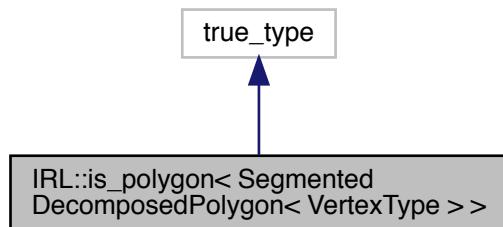
- geometry_type_traits.h

6.616 IRL::is_polygon< SegmentedDecomposedPolygon< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polygon< SegmentedDecomposedPolygon< VertexType > >:



Collaboration diagram for IRL::is_polygon< SegmentedDecomposedPolygon< VertexType > >:

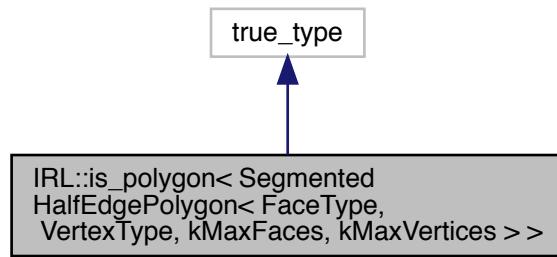


The documentation for this struct was generated from the following file:

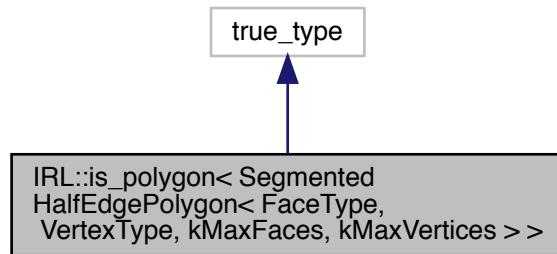
- geometry_type_traits.h

6.617 IRL::is_polygon< SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices >> Struct Template Reference

Inheritance diagram for IRL::is_polygon< SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices >>:



Collaboration diagram for IRL::is_polygon< SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices >>:

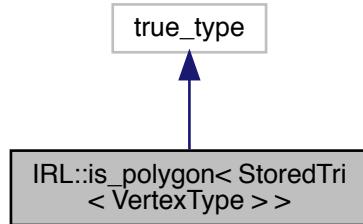


The documentation for this struct was generated from the following file:

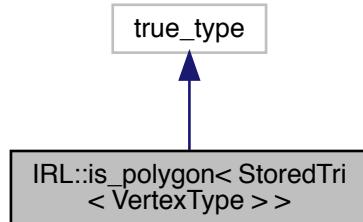
- geometry_type_traits.h

6.618 IRL::is_polygon< StoredTri< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polygon< StoredTri< VertexType > >:



Collaboration diagram for IRL::is_polygon< StoredTri< VertexType > >:

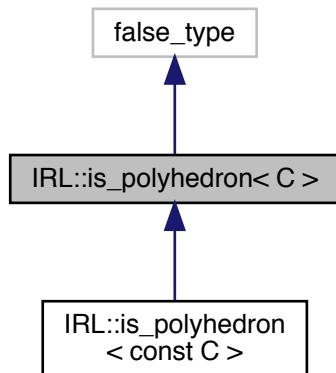


The documentation for this struct was generated from the following file:

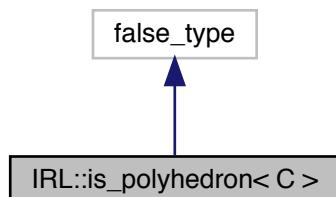
- geometry_type_traits.h

6.619 IRL::is_polyhedron< C > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< C >:



Collaboration diagram for IRL::is_polyhedron< C >:

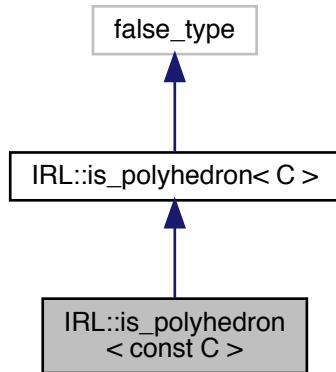


The documentation for this struct was generated from the following file:

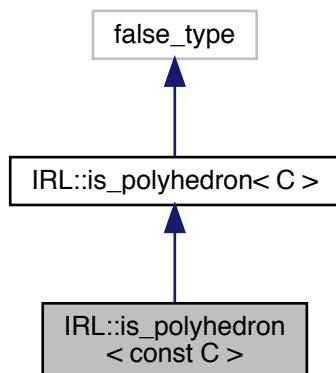
- geometry_type_traits.h

6.620 IRL::is_polyhedron< const C > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< const C >:



Collaboration diagram for IRL::is_polyhedron< const C >:

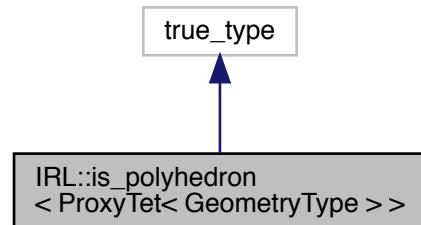


The documentation for this struct was generated from the following file:

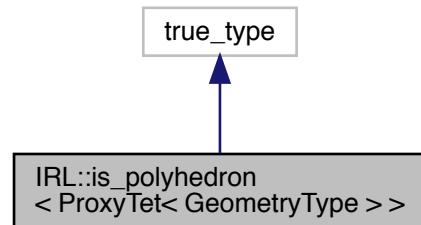
- geometry_type_traits.h

6.621 IRL::is_polyhedron< ProxyTet< GeometryType > > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< ProxyTet< GeometryType > >:



Collaboration diagram for IRL::is_polyhedron< ProxyTet< GeometryType > >:



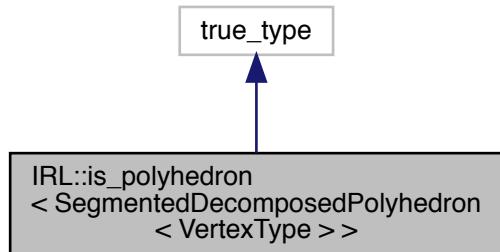
The documentation for this struct was generated from the following file:

- geometry_type_traits.h

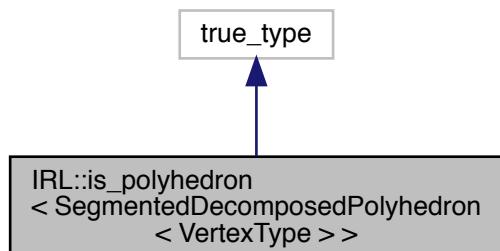
6.622 IRL::is_polyhedron< SegmentedDecomposedPolyhedron< VertexType > >

Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< SegmentedDecomposedPolyhedron< VertexType > >:



Collaboration diagram for IRL::is_polyhedron< SegmentedDecomposedPolyhedron< VertexType > >:

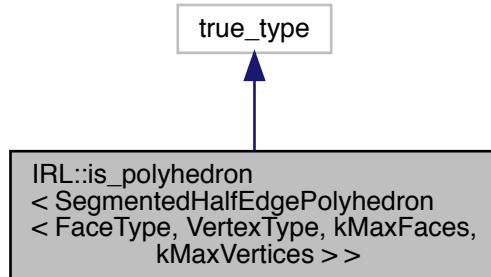


The documentation for this struct was generated from the following file:

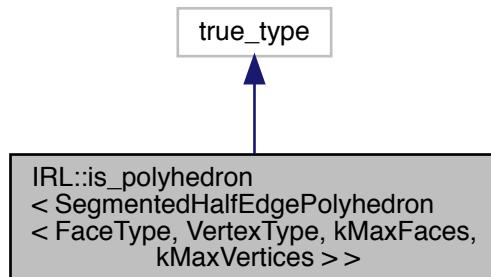
- geometry_type_traits.h

6.623 IRL::is_polyhedron< SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices > > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices > >:



Collaboration diagram for IRL::is_polyhedron< SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices > >:

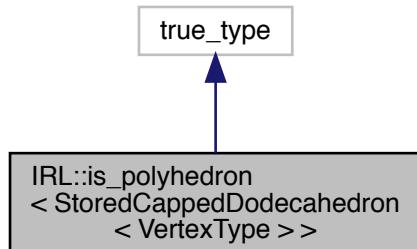


The documentation for this struct was generated from the following file:

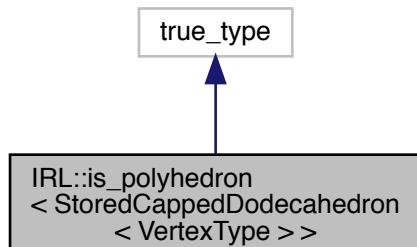
- geometry_type_traits.h

6.624 IRL::is_polyhedron< StoredCappedDodecahedron< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< StoredCappedDodecahedron< VertexType > >:



Collaboration diagram for IRL::is_polyhedron< StoredCappedDodecahedron< VertexType > >:

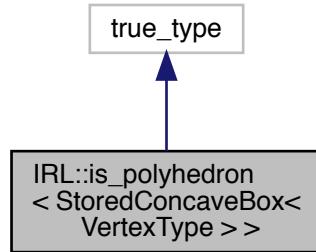


The documentation for this struct was generated from the following file:

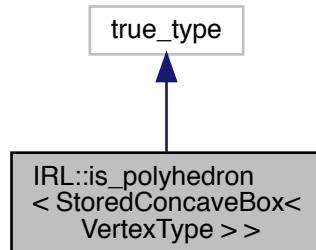
- geometry_type_traits.h

6.625 IRL::is_polyhedron< StoredConcaveBox< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< StoredConcaveBox< VertexType > >:



Collaboration diagram for IRL::is_polyhedron< StoredConcaveBox< VertexType > >:

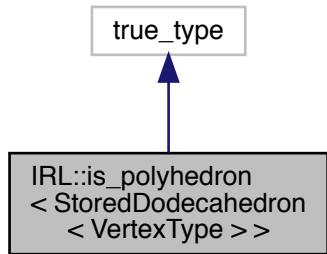


The documentation for this struct was generated from the following file:

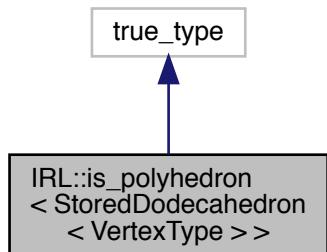
- geometry_type_traits.h

6.626 IRL::is_polyhedron< StoredDodecahedron< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< StoredDodecahedron< VertexType > >:



Collaboration diagram for IRL::is_polyhedron< StoredDodecahedron< VertexType > >:

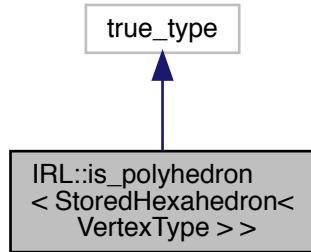


The documentation for this struct was generated from the following file:

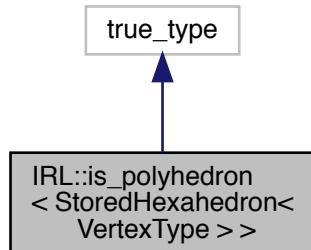
- geometry_type_traits.h

6.627 IRL::is_polyhedron< StoredHexahedron< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< StoredHexahedron< VertexType > >:



Collaboration diagram for IRL::is_polyhedron< StoredHexahedron< VertexType > >:

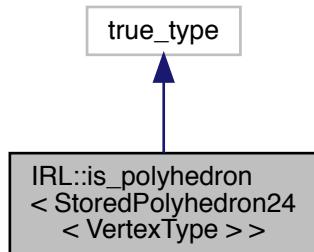


The documentation for this struct was generated from the following file:

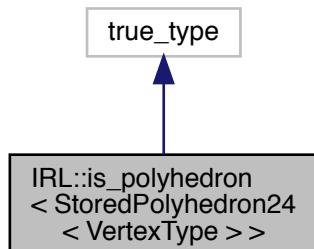
- geometry_type_traits.h

6.628 IRL::is_polyhedron< StoredPolyhedron24< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< StoredPolyhedron24< VertexType > >:



Collaboration diagram for IRL::is_polyhedron< StoredPolyhedron24< VertexType > >:

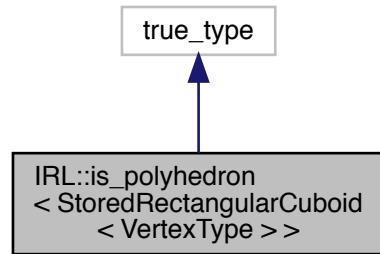


The documentation for this struct was generated from the following file:

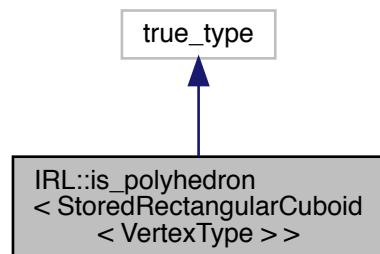
- geometry_type_traits.h

6.629 IRL::is_polyhedron< StoredRectangularCuboid< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< StoredRectangularCuboid< VertexType > >:



Collaboration diagram for IRL::is_polyhedron< StoredRectangularCuboid< VertexType > >:

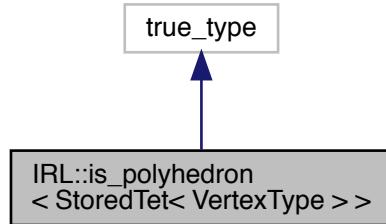


The documentation for this struct was generated from the following file:

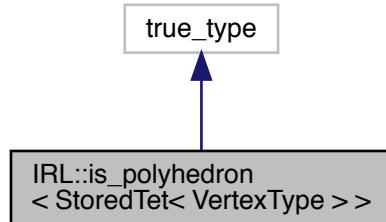
- geometry_type_traits.h

6.630 IRL::is_polyhedron< StoredTet< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_polyhedron< StoredTet< VertexType > >:



Collaboration diagram for IRL::is_polyhedron< StoredTet< VertexType > >:

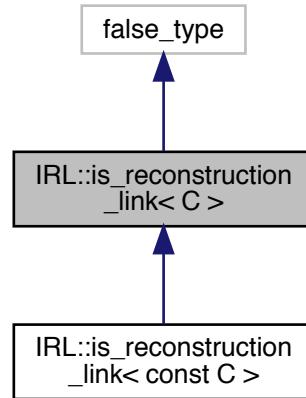


The documentation for this struct was generated from the following file:

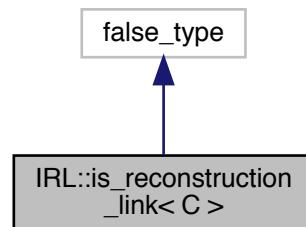
- geometry_type_traits.h

6.631 IRL::is_reconstruction_link< C > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< C >:



Collaboration diagram for IRL::is_reconstruction_link< C >:

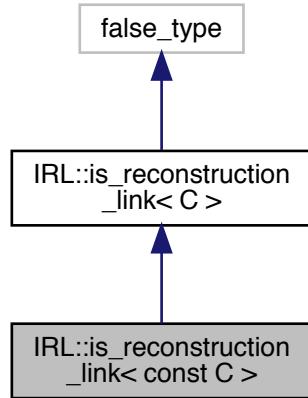


The documentation for this struct was generated from the following file:

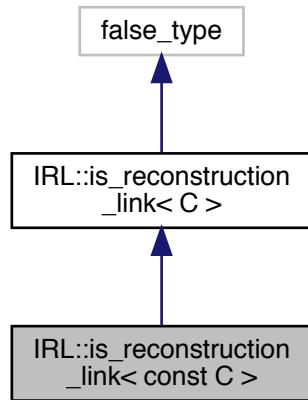
- planar_reconstruction_type_traits.h

6.632 IRL::is_reconstruction_link< const C > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< const C >:



Collaboration diagram for IRL::is_reconstruction_link< const C >:

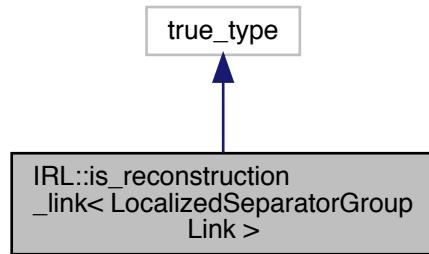


The documentation for this struct was generated from the following file:

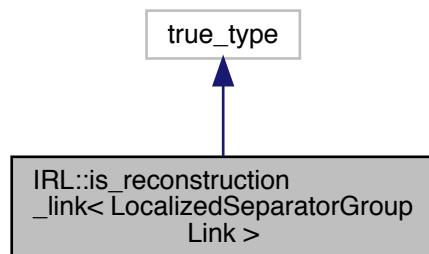
- planar_reconstruction_type_traits.h

6.633 IRL::is_reconstruction_link< LocalizedSeparatorGroupLink > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< LocalizedSeparatorGroupLink >:



Collaboration diagram for IRL::is_reconstruction_link< LocalizedSeparatorGroupLink >:

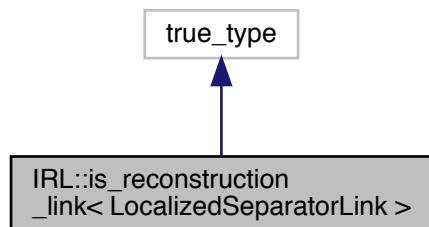


The documentation for this struct was generated from the following file:

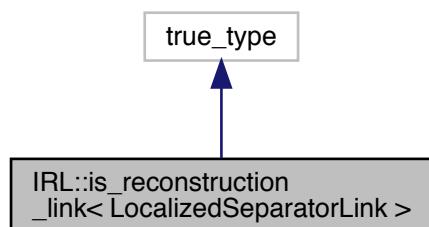
- planar_reconstruction_type_traits.h

6.634 IRL::is_reconstruction_link< LocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< LocalizedSeparatorLink >:



Collaboration diagram for IRL::is_reconstruction_link< LocalizedSeparatorLink >:

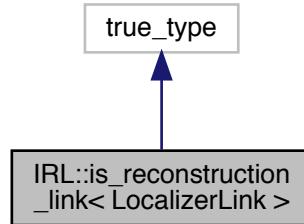


The documentation for this struct was generated from the following file:

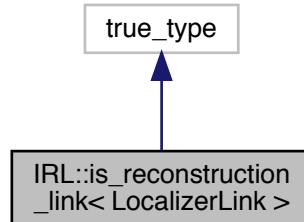
- planar_reconstruction_type_traits.h

6.635 IRL::is_reconstruction_link< LocalizerLink > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< LocalizerLink >:



Collaboration diagram for IRL::is_reconstruction_link< LocalizerLink >:

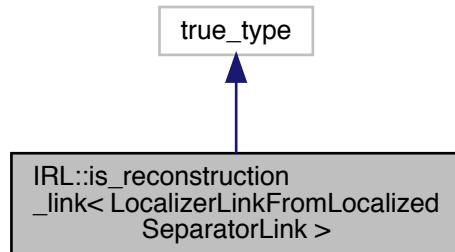


The documentation for this struct was generated from the following file:

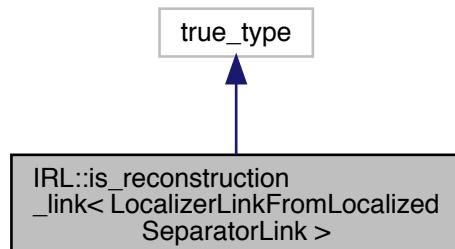
- planar_reconstruction_type_traits.h

6.636 IRL::is_reconstruction_link< LocalizerLinkFromLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< LocalizerLinkFromLocalizedSeparatorLink >:



Collaboration diagram for IRL::is_reconstruction_link< LocalizerLinkFromLocalizedSeparatorLink >:

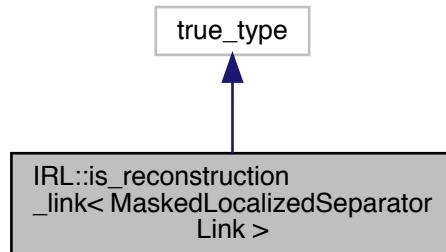


The documentation for this struct was generated from the following file:

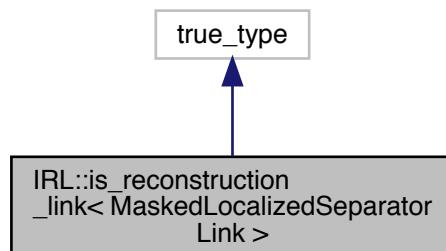
- planar_reconstruction_type_traits.h

6.637 IRL::is_reconstruction_link< MaskedLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< MaskedLocalizedSeparatorLink >:



Collaboration diagram for IRL::is_reconstruction_link< MaskedLocalizedSeparatorLink >:

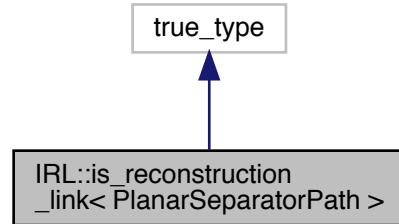


The documentation for this struct was generated from the following file:

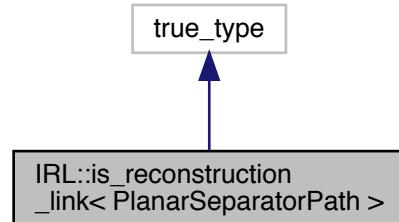
- planar_reconstruction_type_traits.h

6.638 IRL::is_reconstruction_link< PlanarSeparatorPath > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< PlanarSeparatorPath >:



Collaboration diagram for IRL::is_reconstruction_link< PlanarSeparatorPath >:

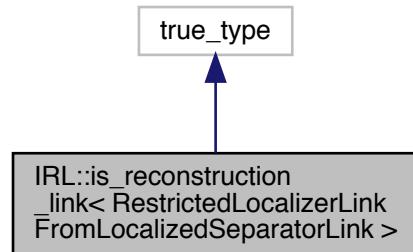


The documentation for this struct was generated from the following file:

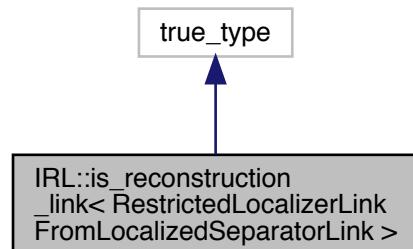
- planar_reconstruction_type_traits.h

6.639 IRL::is_reconstruction_link< RestrictedLocalizerLinkFromLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< RestrictedLocalizerLinkFromLocalizedSeparatorLink >:



Collaboration diagram for IRL::is_reconstruction_link< RestrictedLocalizerLinkFromLocalizedSeparatorLink >:

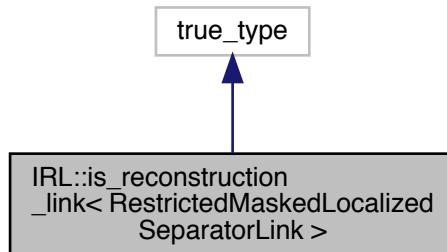


The documentation for this struct was generated from the following file:

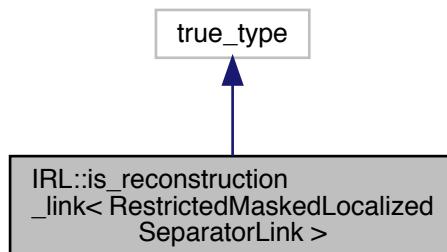
- planar_reconstruction_type_traits.h

6.640 IRL::is_reconstruction_link< RestrictedMaskedLocalizedSeparatorLink > Struct Template Reference

Inheritance diagram for IRL::is_reconstruction_link< RestrictedMaskedLocalizedSeparatorLink >:



Collaboration diagram for IRL::is_reconstruction_link< RestrictedMaskedLocalizedSeparatorLink >:



The documentation for this struct was generated from the following file:

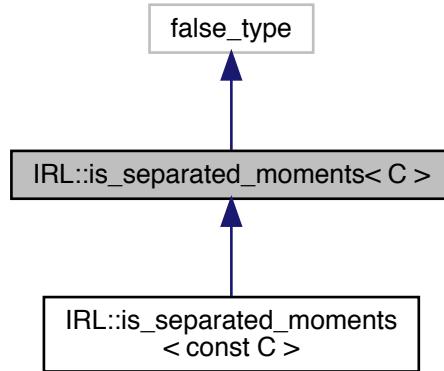
- planar_reconstruction_type_traits.h

6.641 IRL::is_separated_moments< C > Struct Template Reference

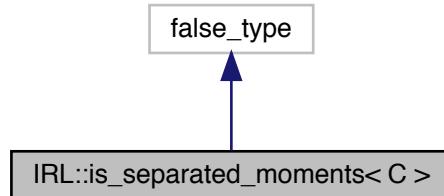
Marking of if volume.

```
#include <moments_type_traits.h>
```

Inheritance diagram for IRL::is_separated_moments< C >:



Collaboration diagram for IRL::is_separated_moments< C >:



6.641.1 Detailed Description

```
template<class C>
struct IRL::is_separated_moments< C >
```

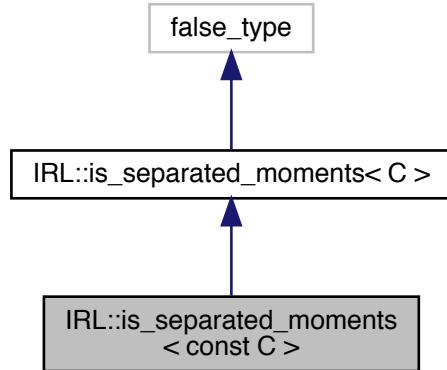
Marking of if volume.

The documentation for this struct was generated from the following file:

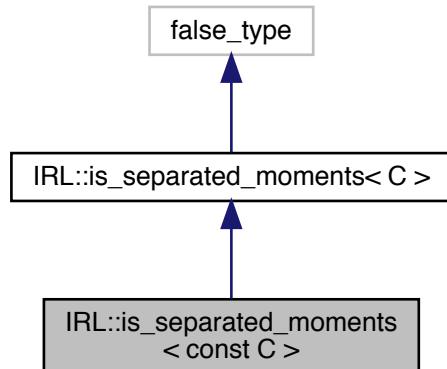
- moments_type_traits.h

6.642 IRL::is_separated_moments< const C > Struct Template Reference

Inheritance diagram for IRL::is_separated_moments< const C >:



Collaboration diagram for IRL::is_separated_moments< const C >:

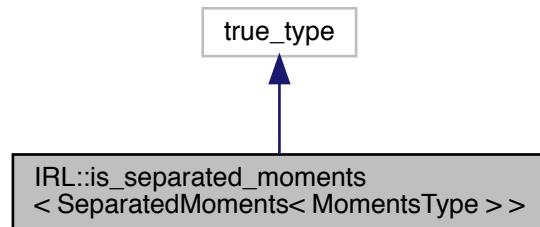


The documentation for this struct was generated from the following file:

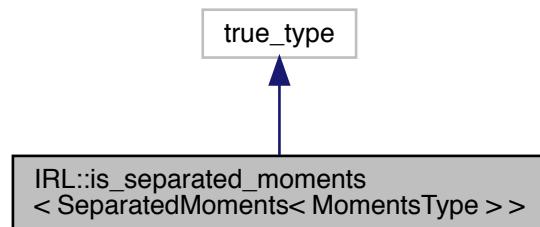
- moments_type_traits.h

6.643 IRL::is_separated_moments< SeparatedMoments< MomentsType > > Struct Template Reference

Inheritance diagram for IRL::is_separated_moments< SeparatedMoments< MomentsType > >:



Collaboration diagram for IRL::is_separated_moments< SeparatedMoments< MomentsType > >:

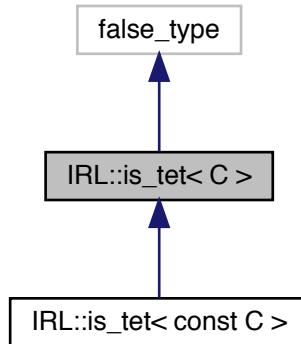


The documentation for this struct was generated from the following file:

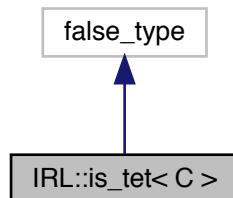
- moments_type_traits.h

6.644 IRL::is_tet< C > Struct Template Reference

Inheritance diagram for IRL::is_tet< C >:



Collaboration diagram for IRL::is_tet< C >:

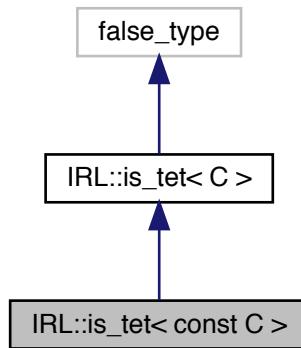


The documentation for this struct was generated from the following file:

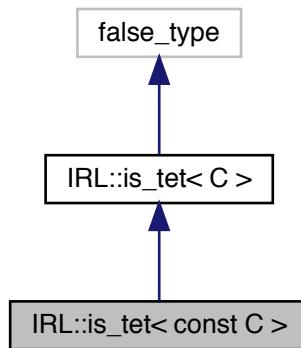
- geometry_type_traits.h

6.645 IRL::is_tet< const C > Struct Template Reference

Inheritance diagram for IRL::is_tet< const C >:



Collaboration diagram for IRL::is_tet< const C >:

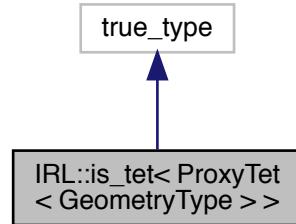


The documentation for this struct was generated from the following file:

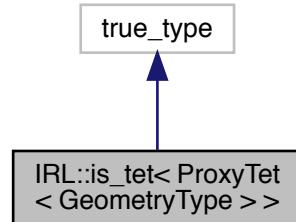
- geometry_type_traits.h

6.646 IRL::is_tet< ProxyTet< GeometryType > > Struct Template Reference

Inheritance diagram for IRL::is_tet< ProxyTet< GeometryType > >:



Collaboration diagram for IRL::is_tet< ProxyTet< GeometryType > >:

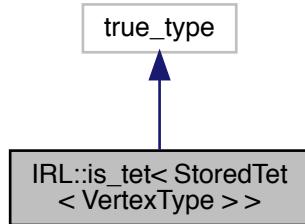


The documentation for this struct was generated from the following file:

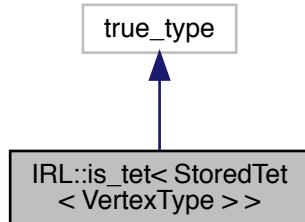
- geometry_type_traits.h

6.647 IRL::is_tet< StoredTet< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_tet< StoredTet< VertexType > >:



Collaboration diagram for IRL::is_tet< StoredTet< VertexType > >:

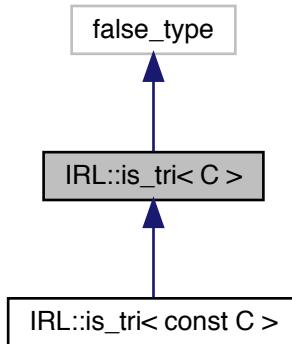


The documentation for this struct was generated from the following file:

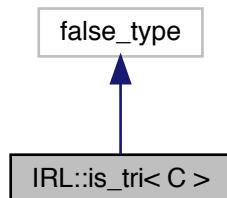
- geometry_type_traits.h

6.648 IRL::is_tri< C > Struct Template Reference

Inheritance diagram for IRL::is_tri< C >:



Collaboration diagram for IRL::is_tri< C >:

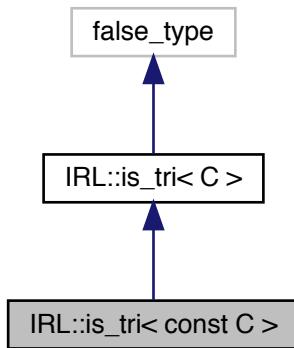


The documentation for this struct was generated from the following file:

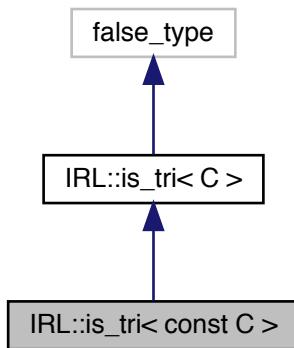
- geometry_type_traits.h

6.649 IRL::is_tri< const C > Struct Template Reference

Inheritance diagram for IRL::is_tri< const C >:



Collaboration diagram for IRL::is_tri< const C >:

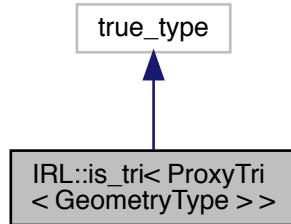


The documentation for this struct was generated from the following file:

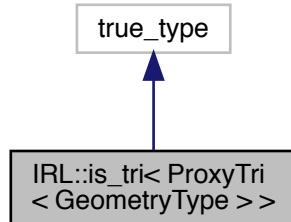
- geometry_type_traits.h

6.650 IRL::is_tri< ProxyTri< GeometryType > > Struct Template Reference

Inheritance diagram for IRL::is_tri< ProxyTri< GeometryType > >:



Collaboration diagram for IRL::is_tri< ProxyTri< GeometryType > >:

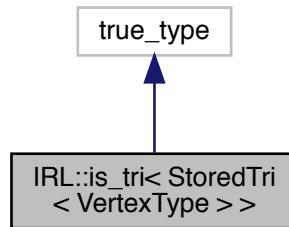


The documentation for this struct was generated from the following file:

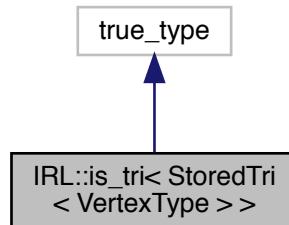
- geometry_type_traits.h

6.651 IRL::is_tri< StoredTri< VertexType > > Struct Template Reference

Inheritance diagram for IRL::is_tri< StoredTri< VertexType > >:



Collaboration diagram for IRL::is_tri< StoredTri< VertexType > >:



The documentation for this struct was generated from the following file:

- geometry_type_traits.h

6.652 f_planarseparator_class::isflipped Interface Reference

Public Member Functions

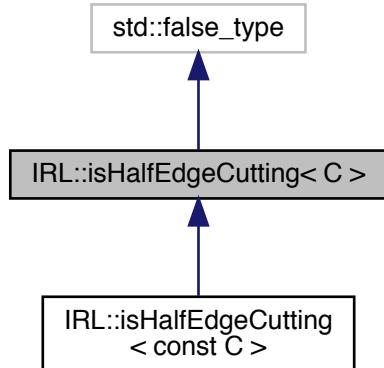
- logical(1) function **planarseparator_class_isflipped** (this)

The documentation for this interface was generated from the following file:

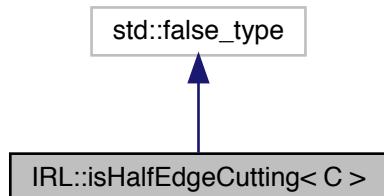
- [f_planarseparator_class.f90](#)

6.653 IRL::isHalfEdgeCutting< C > Struct Template Reference

Inheritance diagram for IRL::isHalfEdgeCutting< C >:



Collaboration diagram for IRL::isHalfEdgeCutting< C >:

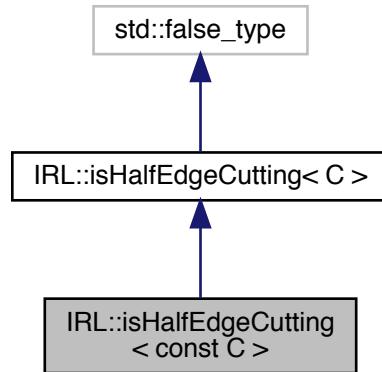


The documentation for this struct was generated from the following file:

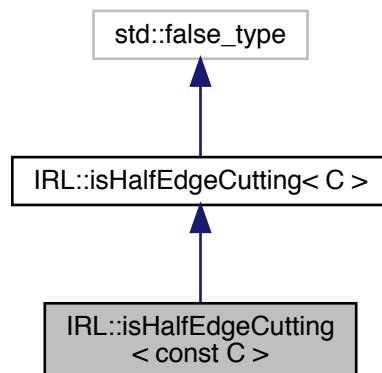
- class_classifications.h

6.654 IRL::isHalfEdgeCutting< const C > Struct Template Reference

Inheritance diagram for IRL::isHalfEdgeCutting< const C >:



Collaboration diagram for IRL::isHalfEdgeCutting< const C >:

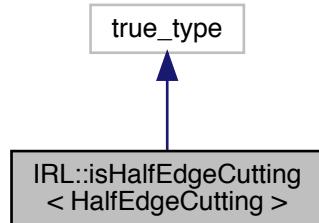


The documentation for this struct was generated from the following file:

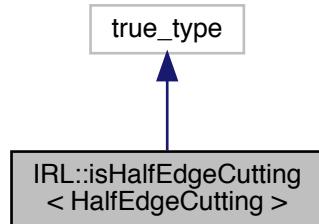
- class_classifications.h

6.655 IRL::isHalfEdgeCutting< HalfEdgeCutting > Struct Template Reference

Inheritance diagram for IRL::isHalfEdgeCutting< HalfEdgeCutting >:



Collaboration diagram for IRL::isHalfEdgeCutting< HalfEdgeCutting >:



The documentation for this struct was generated from the following file:

- class_classifications.h

6.656 IRL::IsNotNullReconstruction< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value**

6.656.1 Member Data Documentation

6.656.1.1 value

```
template<class ReconstructionType >
constexpr bool IRL:: IsNotAPlanarSeparator< ReconstructionType >::value [static]
```

Initial value:

```
= !IsPlanarSeparator<ReconstructionType>::value
```

The documentation for this struct was generated from the following file:

- class_classifications.h

6.657 IRL::IsNotAPlanarSeparator< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value** = !IsPlanarSeparator<ReconstructionType>::value

The documentation for this struct was generated from the following file:

- class_classifications.h

6.658 IRL::IsNullReconstruction< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value**

6.658.1 Member Data Documentation

6.658.1.1 value

```
template<class ReconstructionType >
constexpr bool IRL::IsNullReconstruction< ReconstructionType >::value [static]
```

Initial value:

```
= has_null_reconstruction<ReconstructionType>::value
```

The documentation for this struct was generated from the following file:

- class_classifications.h

6.659 IRL::IsPlanarSeparator< ReconstructionType > Struct Template Reference

Static Public Attributes

- static constexpr bool **value** = [is_planar_separator<ReconstructionType>::value](#)

The documentation for this struct was generated from the following file:

- [class_classifications.h](#)

6.660 f_geometriccuttinghelpers::isptinternal Interface Reference

Public Member Functions

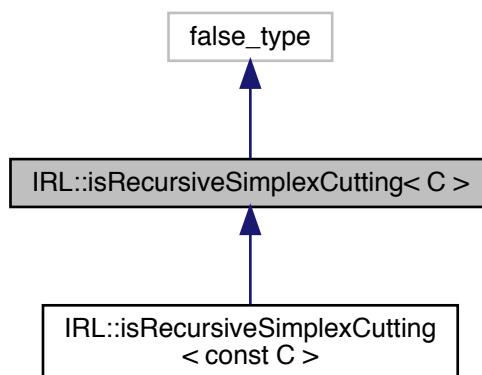
- logical(1) function **isptinternal_ps** (a_pt, a_separator)
- logical(1) function **isptinternal_pl** (a_pt, a_localizer)

The documentation for this interface was generated from the following file:

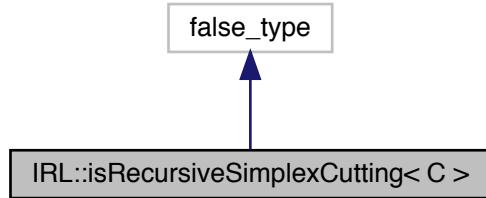
- [f_geometriccuttinghelpers.f90](#)

6.661 IRL::isRecursiveSimplexCutting< C > Struct Template Reference

Inheritance diagram for IRL::isRecursiveSimplexCutting< C >:



Collaboration diagram for IRL::isRecursiveSimplexCutting< C >:

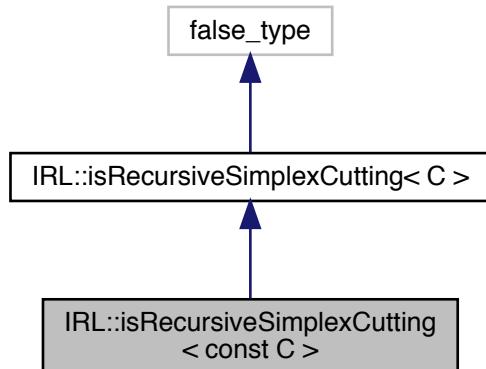


The documentation for this struct was generated from the following file:

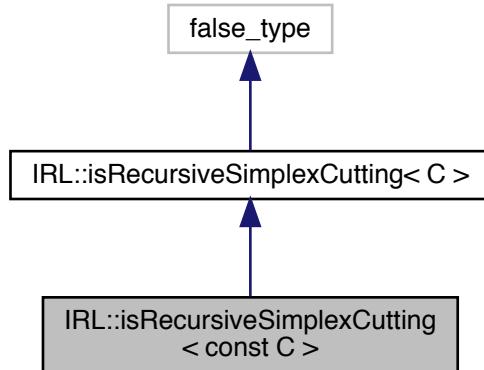
- class_classifications.h

6.662 IRL::isRecursiveSimplexCutting< const C > Struct Template Reference

Inheritance diagram for IRL::isRecursiveSimplexCutting< const C >:



Collaboration diagram for IRL::isRecursiveSimplexCutting< const C >:

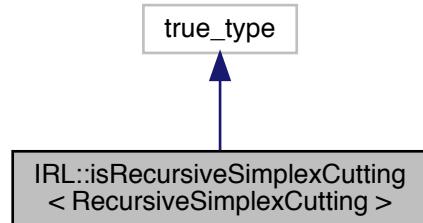


The documentation for this struct was generated from the following file:

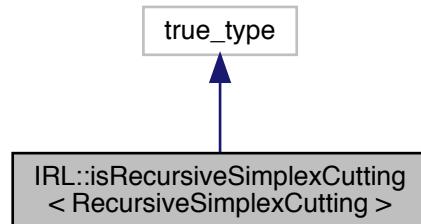
- class_classifications.h

6.663 IRL::isRecursiveSimplexCutting< RecursiveSimplexCutting > Struct Template Reference

Inheritance diagram for IRL::isRecursiveSimplexCutting< RecursiveSimplexCutting >:



Collaboration diagram for IRL::isRecursiveSimplexCutting< RecursiveSimplexCutting >:

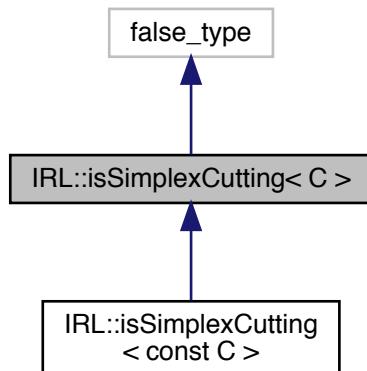


The documentation for this struct was generated from the following file:

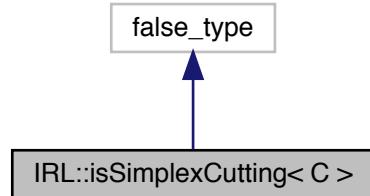
- class_classifications.h

6.664 IRL::isSimplexCutting< C > Struct Template Reference

Inheritance diagram for IRL::isSimplexCutting< C >:



Collaboration diagram for IRL::isSimplexCutting< C >:

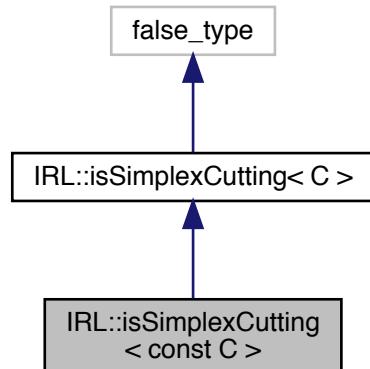


The documentation for this struct was generated from the following file:

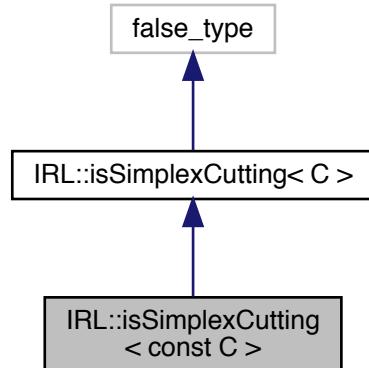
- class_classifications.h

6.665 IRL::isSimplexCutting< const C > Struct Template Reference

Inheritance diagram for IRL::isSimplexCutting< const C >:



Collaboration diagram for IRL::isSimplexCutting< const C >:

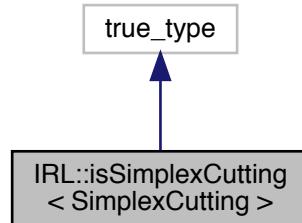


The documentation for this struct was generated from the following file:

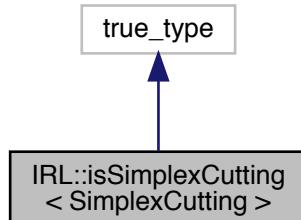
- class_classifications.h

6.666 IRL::isSimplexCutting< SimplexCutting > Struct Template Reference

Inheritance diagram for IRL::isSimplexCutting< SimplexCutting >:



Collaboration diagram for IRL::isSimplexCutting< SimplexCutting >:



The documentation for this struct was generated from the following file:

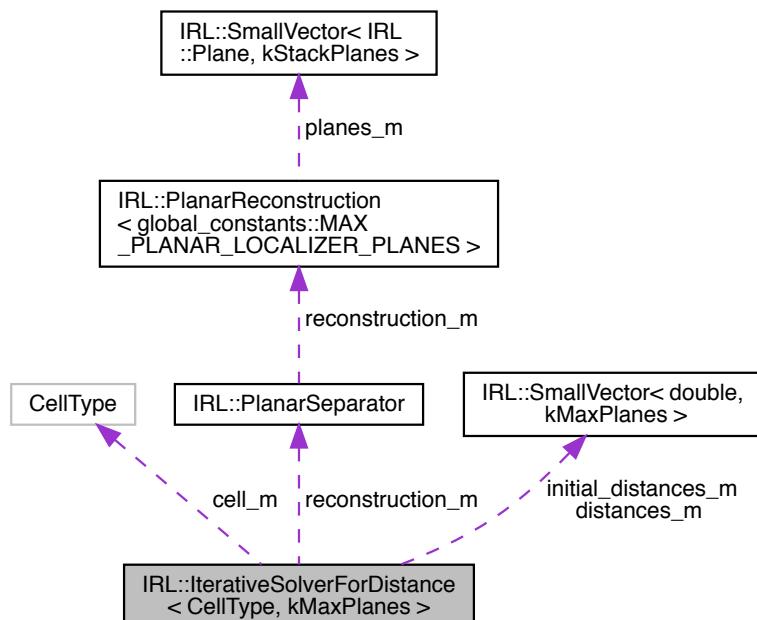
- class_classifications.h

6.667 IRL::IterativeSolverForDistance< CellType, kMaxPlanes > Class Template Reference

[Volume](#) conserving distance-finding routine for two-plane reconstructions wrapped in a class.

```
#include <iterative_distance_solver.h>
```

Collaboration diagram for IRL::IterativeSolverForDistance< CellType, kMaxPlanes >:



Public Member Functions

- **IterativeSolverForDistance** (void)=default
Constructor that initializes the class for optimization and solves for the distance.
- **IterativeSolverForDistance** (const CellType &a_cell, const double a_volume_fraction, const double a_volume_fraction_tolerance, const PlanarSeparator &a_reconstruction)
Constructor that loads in the necessary information to the class and then solves for volume-conserving distance to two planes.
- void **solve** (const CellType &a_cell, const double a_volume_fraction, const double a_volume_fraction_tolerance, const PlanarSeparator &a_reconstruction)
Reinitialize solver and solve for distance to each plane.
- void **updateGuess** (double *a_delta)
Update the current guess and solution given the delta change.
- void **setGuess** (double *a_guess)
Set new guess distances to each plane.
- double **calculateSignedScalarError** (void)
Calculate current volume fraction for reconstruction_m with the plane distances_m, return error.
- bool **errorTooHigh** (const double a_error)
Return bool for whether the error is still too high.
- bool **iterationTooHigh** (const UnsignedIndex_t a_iteration)
Return bool for whether maximum iterations has been exceeded.
- void **updateBoundaries** (const double a_error)
Update known boundaries for solution.
- SmallVector< double, kMaxPlanes > **getDistances** (void)
Return pointer to distances_m to be used for changing a reconstruction.
- double **getDistances** (const UnsignedIndex_t a_p)
Return the distance for plane a_p.
- ~**IterativeSolverForDistance** (void)=default
Default destructor.

Private Member Functions

- void **setup** (void)
Prepare the object to be passed to generalized solvers.
- void **solveForDistance** (void)
Main driving function that solves for distances.
- bool **isBoundsTrueBounds** (void)
Make sure initial bounds are set up correctly.
- void **checkIfStaticAllocationExceeded** (void) const

Private Attributes

- CellType **cell_m**
Cell distance is being calculated for.
- double **target_volume_fraction_m**
Target volume fraction to match.
- double **volume_fraction_tolerance_m**
Tolerance to match volume fraction within.
- PlanarSeparator **reconstruction_m**
The reconstruction the distance is being found for.

- `SmallVector< double, kMaxPlanes > initial_distances_m`
Clipped initial distance that was in reconstruction_m that the optimization will start from.
- `SmallVector< double, kMaxPlanes > distances_m`
Distances that the planes in reconstruction should be set to.
- `double current_guess_m`
The current solution.
- `std::array< double, 3 > bound_value_m`
Values bracketing solution.
- `std::array< double, 3 > bound_error_m`
Errors for values bracketing solution.

Static Private Attributes

- `static constexpr UnsignedIndex_t max_iter_m = {15}`
Max number of iterations for the Newton-Raphson Solver.
- `static constexpr UnsignedIndex_t max_bisection_iter = {40}`

6.667.1 Detailed Description

```
template<class CellType, UnsignedIndex_t kMaxPlanes = global_constants::MAX_PLANAR_SEPARATOR_PLANES>
class IRL::IterativeSolverForDistance< CellType, kMaxPlanes >
```

`Volume` conserving distance-finding routine for two-plane reconstructions wrapped in a class.

This class finds the distance to the two planes provided in `a_reconstruction` to recreate `a_volume_fraction` within the tolerance `a_find_volume_tolerance`. This is first attempted by a Newton-Raphson optimization. If the optimization does not reach within the tolerance in a given number of iterations, it then defaults to using bisection with the boundaries being the current best solutions on both sides of the goal `volume_fraction`.

Parameters

in	<code>volume_fraction_m</code>	<code>Volume</code> fraction to recreate
in	<code>volume_fraction_tolerance_m</code>	Tolerance to recreate <code>a_volume_fraction</code> within
in	<code>reconstruction_m</code>	A copy of the reconstruction to find distances for. param[out] <code>distances_m</code> Correct distance to plane is stored in <code>distances_m</code> after construction of the object.

6.667.2 Constructor & Destructor Documentation

6.667.2.1 IterativeSolverForDistance() [1/2]

```
template<class CellType , UnsignedIndex_t kMaxPlanes = global_constants::MAX_PLANAR_SEPARATOR_PLANES>
```

```
IRL::IterativeSolverForDistance< CellType, kMaxPlanes >::IterativeSolverForDistance (
    void ) [default]
```

Constructor that initializes the class for optimization and solves for the distance.

Default constructor

6.667.2.2 IterativeSolverForDistance() [2/2]

```
template<class CellType , UnsignedIndex_t kMaxPlanes = global_constants::MAX_PLANAR_SEPARATOR_PLACES>
IRL::IterativeSolverForDistance< CellType, kMaxPlanes >::IterativeSolverForDistance (
    const CellType & a_cell,
    const double a_volume_fraction,
    const double a_volume_fraction_tolerance,
    const PlanarSeparator & a_reconstruction )
```

Constructor that loads in the neccessary information to the class and then solves for volume-conserving distance to two planes.

Parameters

in	a_volume_fraction	Volume fraction to match.
in	a_volume_fraction_tolerance	Tolerance to recreate volume fraction within.
in	a_reconstruction	Reconstruction to find distances for.

6.667.3 Member Function Documentation

6.667.3.1 solve()

```
template<class CellType , UnsignedIndex_t kMaxPlanes = global_constants::MAX_PLANAR_SEPARATOR_PLACES>
void IRL::IterativeSolverForDistance< CellType, kMaxPlanes >::solve (
    const CellType & a_cell,
    const double a_volume_fraction,
    const double a_volume_fraction_tolerance,
    const PlanarSeparator & a_reconstruction )
```

Reinitialize solver and solve for distance to each plane.

Parameters

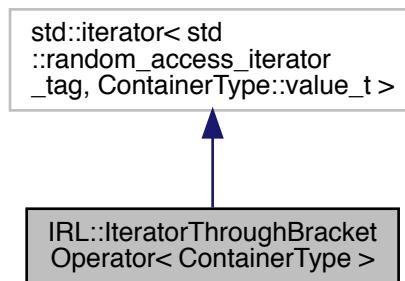
in	a_volume_fraction	Volume fraction to match.
in	a_volume_fraction_tolerance	Tolerance to recreate volume fraction within.
in	a_reconstruction	Reconstruction to find distances for.

The documentation for this class was generated from the following file:

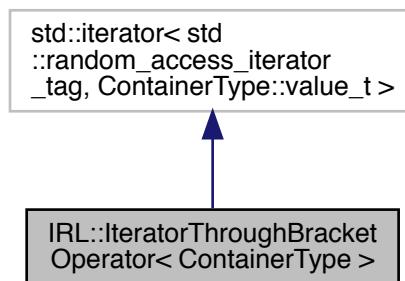
- iterative_distance_solver.h

6.668 IRL::IteratorThroughBracketOperator< ContainerType > Class Template Reference

Inheritance diagram for IRL::IteratorThroughBracketOperator< ContainerType >:



Collaboration diagram for IRL::IteratorThroughBracketOperator< ContainerType >:



Public Types

- using **value_t** = typename ContainerType::value_t

Public Member Functions

- **IteratorThroughBracketOperator** (ContainerType &a_container, const UnsignedIndex_t a_location)
- value_t & **operator*** (void)
- value_t & **operator->** (void)
- value_t & **operator[]** (const std::ptrdiff_t a_index)
- **IteratorThroughBracketOperator** & **operator++** (void)
- **IteratorThroughBracketOperator** & **operator--** (void)
- **IteratorThroughBracketOperator** **operator++** (int a_dummy_for_postfix)
- **IteratorThroughBracketOperator** **operator--** (int a_dummy_for_postfix)
- **IteratorThroughBracketOperator** & **operator+=** (const std::ptrdiff_t a_shift)
- **IteratorThroughBracketOperator** & **operator-=** (const std::ptrdiff_t a_shift)
- std::ptrdiff_t **operator-** (const **IteratorThroughBracketOperator** &a_rhs)
- bool **operator==** (const **IteratorThroughBracketOperator** &a_rhs) const
- bool **operator!=** (const **IteratorThroughBracketOperator** &a_rhs) const
- bool **operator<** (const **IteratorThroughBracketOperator** &a_rhs) const
- bool **operator>** (const **IteratorThroughBracketOperator** &a_rhs) const
- bool **operator>=** (const **IteratorThroughBracketOperator** &a_rhs) const
- bool **operator<=** (const **IteratorThroughBracketOperator** &a_rhs) const

Private Attributes

- ContainerType & **container_m**
- UnsignedIndex_t **location_m**

The documentation for this class was generated from the following file:

- iterator_through_bracket_operator.h

6.669 IRL::JoinedReconstructions< CurrentReconstruction, NextReconstruction > Class Template Reference

Class that ties together two reconstructions. Will first cut by and use the current reconstruction, and then pass onwards to the next reconstruction.

```
#include <joined_reconstructions.h>
```

Public Member Functions

- **JoinedReconstructions** (void)
Default constructor.
- **JoinedReconstructions** (const CurrentReconstruction *a_current_reconstruction_ptr, const NextReconstruction *a_next_reconstruction_ptr)
Construct with providing a pointer to a localizer and separator.
- const CurrentReconstruction & **getFirstReconstruction** (void) const
Return the current reconstruction in JoinedReconstructions.
- const NextReconstruction & **getSecondReconstruction** (void) const
Return the next reconstruction in the JoinedReconstructions.
- const auto & **getCurrentReconstruction** (void) const
Return the current reconstruction in JoinedReconstructions.
- const NextReconstruction & **getNextReconstruction** (void) const
Return the next reconstruction in the JoinedReconstructions.
- **~JoinedReconstructions** (void)=default
Default destructor;

Private Attributes

- const CurrentReconstruction * **current_reconstruction_m**
- const NextReconstruction * **next_reconstruction_m**

6.669.1 Detailed Description

```
template<class CurrentReconstruction, class NextReconstruction>
class IRL::JoinedReconstructions< CurrentReconstruction, NextReconstruction >
```

Class that ties together two reconstructions. Will first cut by and use the current reconstruction, and then pass onwards to the next reconstruction.

The documentation for this class was generated from the following file:

- joined_reconstructions.h

6.670 IRL::JoinedReconstructions< CurrentReconstruction, NullReconstruction > Class Template Reference

Public Member Functions

- [JoinedReconstructions](#) (void)
Default constructor.
- [JoinedReconstructions](#) (const CurrentReconstruction **a_current_reconstruction_ptr*)
Construct with providing a pointer to a localizer and separator.
- const CurrentReconstruction & [getFirstReconstruction](#) (void) const
Return the first reconstruction (CurrentReconstruction)
- [NullReconstruction](#) [getSecondReconstruction](#) (void) const
Return the second reconstruction (NextReconstruction)
- const auto & [getCurrentReconstruction](#) (void) const
Return the current reconstruction of the first reconstruction.
- [NullReconstruction](#) [getNextReconstruction](#) (void) const
Return the second reconstruction.
- [~JoinedReconstructions](#) (void)=default
Default destructor.;

Private Types

- using **NextReconstruction** = [NullReconstruction](#)

Private Attributes

- const CurrentReconstruction * **current_reconstruction_m**

The documentation for this class was generated from the following file:

- joined_reconstructions.h

6.671 IRL::JoinedReconstructionsToGroup< CurrentReconstruction, GroupReconstruction > Class Template Reference

Public Member Functions

- `JoinedReconstructionsToGroup` (void)
Default constructor.
- `JoinedReconstructionsToGroup` (const CurrentReconstruction *`a_current_reconstruction_ptr`, const GroupReconstruction *`a_next_reconstruction_ptr`)
Construct with providing a pointer to a localizer and separator.
- `const CurrentReconstruction & getFirstReconstruction` (void) const
Return the current reconstruction in `JoinedReconstructions`.
- `const GroupReconstruction & getSecondReconstruction` (void) const
Return the next reconstruction in the `JoinedReconstructions`.
- `const auto & getCurrentReconstruction` (void) const
Return the current reconstruction in `JoinedReconstructions`.
- `const auto & getNextReconstruction` (void) const
Return the next reconstruction in the `JoinedReconstructions`.
- `~JoinedReconstructionsToGroup` (void)=default
Default destructor;

Private Attributes

- `const CurrentReconstruction * current_reconstruction_m`
- `const GroupReconstruction * next_reconstruction_m`

The documentation for this class was generated from the following file:

- `joined_reconstructions_to_group.h`

6.672 IRL::KMeans Struct Reference

A class that executes Kmeans when provided an appropriate DrivingClass.

```
#include <k_means.h>
```

Static Public Member Functions

- `template<class DrivingClass >`
`static UnsignedIndex_t partition` (`DrivingClass *a_driving_ptr`)
Partition into `KMeans` according to `DrivingClass`.

6.672.1 Detailed Description

A class that executes Kmeans when provided an appropriate DrivingClass.

This function is used to drive Kmeans in order to partition a set into multiple sets. This is done in a very broad way which allows it to perform this partitioning and the effect of the partitioning in many ways, dictated by the driving class.

Requirements for DrivingClass:

- `bool isDone(void)` : A method that determines if the `KMeans` routine should be stopped and the initial set has been sufficiently partitioned. -`void setupNextIteration(void)` : A method to prepare everything necessary in the DrivingClass object for the next iteration/partitioning.
- `const_iterator begin()` : Method that returns a const iterator to the start of the initial set being partitioned.
- `const_iterator end()` : Method that returns a const iterator to the end of the initial set being partitioned. -`int findCorrectPartition(const ElementType&)` : A method that takes an element from the initial set in the DrivingClass object and returns the partition in DrivingClass that it should belong to. -`void addElementToPartition(const int, const ElementType&)` : Given the partition the element should belong to (determined by the `findCorrectPartition` method, the element is added to the partition. Can be used to directly update the consequence instead, such as updating a sum instead of directly constructing sets representing the partition. -`bool iterationTooHigh(const UnsignedIndex_t)` : A method that takes the number of iterations and returns a bool whether the maximum number of allowable iterations has been exceeded.

The documentation for this struct was generated from the following file:

- `k_means.h`

6.673 IRL::LevenbergMarquardt< OptimizingClass, kRows, kColumns > Class Template Reference

Levenberg-Marquardt optimization routine.

```
#include <levenberg_marquardt.h>
```

Public Member Functions

- `LevenbergMarquardt(void)`
Default construction.
- `void solve(OptimizingClass *a_setup_otype, const Eigen::Matrix< double, kColumns, 1 > &a_jacobian_<- delta)`
Assigns otype_m to a new pointer and then solves.
- `int getReason(void)`
Return reason for exiting by integer.
- `UnsignedIndex_t getCount(void)`
Return the number of iterations it took until exit.
- `~LevenbergMarquardt(void)=default`
Default dedestructor.

Private Member Functions

- void `solve` (const Eigen::Matrix< double, kColumns, 1 > &a_jacobian_delta)
Perform non-linear optimization.
- void `calculateJacobian` (const Eigen::Matrix< double, kColumns, 1 > &a_delta, Eigen::Matrix< double, kColumns, kRows > *a_jacobian_transpose, Eigen::Matrix< double, kColumns, kColumns > *a_jacTjac)
Calculate jacobian using first-order finite difference.

Private Attributes

- OptimizingClass * `otype_m`
Pointer to object of class OptimizingClass that is being optimized.
- UnsignedIndex_t `iteration_m`
Iterations of Levenberg-Marquardt algorithm.
- int `reason_for_exit_m`
Integer indicating reason for Levenberg-Marquardt exiting.
- Eigen::Matrix< double, kColumns, kRows > `jacobian_transpose_m`
Transpose of Jacobian for guess vector.
- Eigen::Matrix< double, kColumns, kColumns > `jacTjac_m`

$$\text{Matrix } \text{JacTJac_m} = \text{jacobian_transpose_m} * \text{transpose}(\text{jacobian_transpose_m}).$$
- Eigen::Matrix< double, kColumns, 1 > `delta_m`
Change in parameters being fit.
- Eigen::Matrix< double, kColumns, kColumns > `A_m`

$$(\text{JacTJac_m} + \lambda * \text{I}), \text{ and preconditioned with Jacobi pre-conditioner.}$$
- Eigen::Matrix< double, kRows, 1 > `vector_error_m`
Error vector of correct - guess.
- Eigen::Matrix< double, kColumns, 1 > `rhs_m`

$$\text{RHS of } (\text{JacTJac_m} + \lambda * \text{I}) * \text{delta} = \text{jacobian_transpose_m} * (\text{vector_error_m})$$
- Eigen::Matrix< double, kColumns, 1 > `rhs_precond_m`
rhs_m with Jacobi preconditioner applied.

6.673.1 Detailed Description

```
template<class OptimizingClass, int kRows, int kColumns>
class IRL::LevenbergMarquardt< OptimizingClass, kRows, kColumns >
```

Levenberg-Marquardt optimization routine.

Requirements for OptimizingClass:

- double `calculateScalarError(void)` : A method to calculate a scalar error that we are trying to minimize.
- Eigen::Matrix<double, kRows, 1> `calculateVectorError(void)` : A method that returns the vector return (`correct_values - guess_values`) by value
- void `updateGuess(Eigen::Matrix<double, kColumns, 1>)` : A method that takes in the delta change and computes a new guess vector (which it is storing itself)
- void `updateBestGuess(void)` : A method that updates the best guess and all other things necessary before a new Jacobian is calculated and a new step is taken.

- Eigen::Matrix<double, kRows, 1> calculateChangeInGuess (void) : A method that calculates the difference between Guess variables and bestGuess (for use in calculating derivative in Jacobian).
- void increaseLambda(double*) : A method to increase the value of lambda (for failed attempts at finding a new minimum).
- void decreaseLambda(double*) : A method to decrease the value of lambda (for successful attempts at finding a new minimum).
- bool errorTooHigh(const double) : A method that takes a scalar error and returns a boolean whether the error is low enough to stop optimization and return. -bool iterationTooHigh(const int) : A method that takes the number of iterations and returns a bool whether the maximum number of allowable iterations has been exceeded.
- minimumReached(const Eigen::Matrix<double, kColumns, 1>) : A method that takes delta and determines if the optimization has reached a minimum, return a bool true if optimization should exit. shouldComputeJacobian(const int, const int) : A method that returns a bool for whether or not a jacobian should be computed when given the current iteration and the last iteration the Jacobian was computed for.

kRows is the number of rows involved in the error vector of the Levenberg-Marquardt system [y-f].

kColumns is the number of columns involved in the Jacobian for the Levenberg-Marquardt system, equal to the number of parameters being fit.

6.673.2 Member Data Documentation

6.673.2.1 reason_for_exit_m

```
template<class OptimizingClass , int kRows, int kColumns>
int IRL::LevenbergMarquardt< OptimizingClass, kRows, kColumns >::reason_for_exit_m [private]
```

Integer indicating reason for Levenberg-Marquardt exiting.

Reasons:

- ≥ 0 : Number of iterations taken reduce error to acceptable level.
- -1 : Exited due to exceeding maximum number of iterations.
- -2 : Exited due to minimum reached (largest magnitude in delta less than set amount).

The documentation for this class was generated from the following file:

- levenberg_marquardt.h

6.674 IRL::LevenbergMarquardt< OptimizingClass, -1, kColumns > Class Template Reference

Public Member Functions

- [LevenbergMarquardt](#) (void)

Default construction.
- void [solve](#) (OptimizingClass *a_setup_otype, const int a_number_of_rows, const Eigen::Matrix< double, kColumns, 1 > &a_jacobian_delta)

Assigns otype_m to a new pointer and then solves.
- int [getReason](#) (void)

Return reason for exiting by integer.
- UnsignedIndex_t [getIterationCount](#) (void)

Return the number of iterations it took until exit.
- [~LevenbergMarquardt](#) (void)=default

Default destructor.

Private Member Functions

- void [solve](#) (const int a_number_of_rows, const Eigen::Matrix< double, kColumns, 1 > &a_jacobian_delta)

Perform non-linear optimization to get construction according to R2P.
- void [calculateJacobian](#) (const Eigen::Matrix< double, kColumns, 1 > &a_delta, Eigen::Matrix< double, kColumns, Eigen::Dynamic > *a_jacobian_tr transpose, Eigen::Matrix< double, kColumns, kColumns > *a_jacTjac)

Calculate jacobian using first-order finite difference.

Private Attributes

- OptimizingClass * [otype_m](#)

Pointer to object of class OptimizingClass that is being optimized.
- UnsignedIndex_t [iteration_m](#)

Iterations of Levenberg-Marquardt algorithm.
- int [reason_for_exit_m](#)

Integer indicating reason for Levenberg-Marquardt exiting.
- Eigen::Matrix< double, kColumns, Eigen::Dynamic > [jacobian_tr transpose_m](#)

Transpose of Jacobian for guess vector.
- Eigen::Matrix< double, kColumns, kColumns > [jacTjac_m](#)

*Matrix $\text{JacTJac}_m = \text{jacobian_transpose}_m * \text{transpose}(\text{jacobian_transpose}_m)$.*
- Eigen::Matrix< double, kColumns, 1 > [delta_m](#)

Change in parameters being fit.
- Eigen::Matrix< double, kColumns, kColumns > [A_m](#)

*$(\text{JacTJac}_m + \lambda*I)$, and preconditioned with Jacobi pre-conditioner.*
- Eigen::Matrix< double, Eigen::Dynamic, 1 > [vector_error_m](#)

Error vector of correct - guess.
- Eigen::Matrix< double, kColumns, 1 > [rhs_m](#)

*RHS of $(\text{JacTJac}_m + \lambda*I)*\text{delta} = \text{jacobian_transpose}_m * (\text{vector_error}_m)$*
- Eigen::Matrix< double, kColumns, 1 > [rhs_precond_m](#)

rhs_m with Jacobi preconditioner applied.

6.674.1 Member Data Documentation

6.674.1.1 reason_for_exit_m

```
template<class OptimizingClass , int kColumns>
int IRL::LevenbergMarquardt< OptimizingClass, -1, kColumns >::reason_for_exit_m [private]
```

Integer indicating reason for Levenberg-Marquardt exiting.

Reasons:

- ≥ 0 : Number of iterations taken reduce error to acceptable level.
- -1 : Exited due to exceeding maximum number of iterations.
- -2 : Exited due to minimum reached (largest magnitude in delta less than set amount).

The documentation for this class was generated from the following file:

- levenberg_marquardt.h

6.675 IRL::LinearInterpolation_Functor Class Reference

Public Member Functions

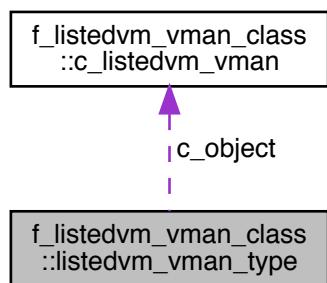
- template<class PtWithDataType >
PtWithDataType **operator()** (const PtWithDataType &a_pt_0, const double a_dist_0, const PtWithDataType &a_pt_1, const double a_dist_1)

The documentation for this class was generated from the following file:

- new_pt_calculation_functors.h

6.676 f_listedvm_vman_class::listedvm_vman_type Type Reference

Collaboration diagram for f_listedvm_vman_class::listedvm_vman_type:



Public Member Functions

- final `listedvm_vman_class_delete`

Private Attributes

- type([c_listedvm_vman](#)), private `c_object`

The documentation for this type was generated from the following file:

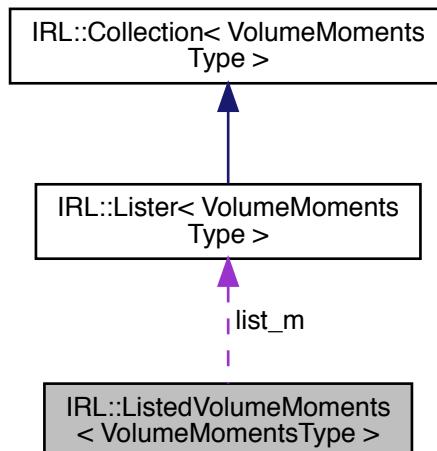
- `f_listedvm_vman_class.f90`

6.677 IRL::ListedVolumeMoments< VolumeMomentsType > Class Template Reference

[VolumeMoments](#) wrapper for [Lister](#) class to add ability to normalize.

```
#include <listed_volume_moments.h>
```

Collaboration diagram for IRL::ListedVolumeMoments< VolumeMomentsType >:



Public Types

- using `contained_type` = `VolumeMomentsType`
- using `iterator` = typename [Lister< VolumeMomentsType >::iterator](#)
- using `const_iterator` = typename [Lister< VolumeMomentsType >::const_iterator](#)

Public Member Functions

- `ListedVolumeMoments` (void)=default
Default constructor.
- `UnsignedIndex_t size` (void) const
Get size of the collection.
- `contained_type & operator[]` (const `UnsignedIndex_t` a_index)
This will self-expand to prevent itself from accessing out of bounds memory.
- `const contained_type & operator[]` (const `UnsignedIndex_t` a_index) const
Const version for access to object in collection.
- `ListedVolumeMoments & operator=` (const double a_value)
Set all moments in the list equal to a_value.
- `void normalizeByVolume` (void)
Normalize entire vector by volume.
- `void multiplyByVolume` (void)
Normalize entire vector by volume.
- `ListedVolumeMoments & operator+=` (const `ListedVolumeMoments` &a_rhs)
Overload operator+= to accumulate moments with the same index, and extend to match a_rhs length.
- `ListedVolumeMoments & operator+=` (const `VolumeMomentsType` &a_rhs)
The operator += will be used to push_back the object a_rhs in the current collection in Lister.
- `void clear` (void)
Empty the container.
- `void erase` (const `UnsignedIndex_t` a_index)
Erase an object from the container at the index.
- iterator `begin` (void) noexcept
- const_iterator `begin` (void) const noexcept
- const_iterator `cbegin` (void) const noexcept
- iterator `end` (void) noexcept
- const_iterator `end` (void) const noexcept
- const_iterator `cend` (void) const noexcept
- `~ListedVolumeMoments` (void)=default
Default destructor.

Private Attributes

- `Lister< VolumeMomentsType > list_m`

6.677.1 Detailed Description

```
template<class VolumeMomentsType>
class IRL::ListedVolumeMoments< VolumeMomentsType >
```

`VolumeMoments` wrapper for `Lister` class to add ability to normalize.

The documentation for this class was generated from the following file:

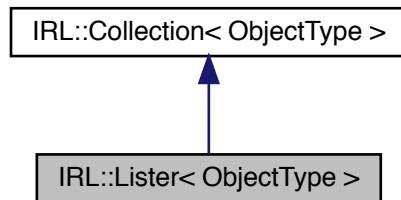
- `listed_volume_moments.h`

6.678 IRL::Lister< ObjectType > Class Template Reference

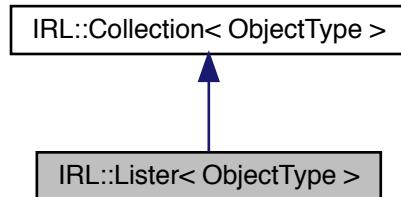
This class is a [Collection](#) that has operator += add the other [Lister](#) or object to the back of its own collection.

```
#include <lister.h>
```

Inheritance diagram for IRL::Lister< ObjectType >:



Collaboration diagram for IRL::Lister< ObjectType >:



Public Member Functions

- [Lister](#) (void)=default
Default constructor.
- [Lister](#) & [operator+=](#) (const [Lister](#) &a_rhs)
The operator += will be used to append a_rhs to the current collection in Lister.
- [Lister](#) & [operator+=](#) (const ObjectType &a_rhs)
The operator += will be used to push_back the object a_rhs in the current collection in Lister.
- [~Lister](#) (void)=default
Default destructor.

Additional Inherited Members

6.678.1 Detailed Description

```
template<class ObjectType>
class IRL::Lister< ObjectType >
```

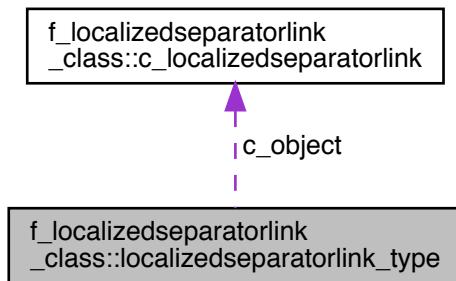
This class is a [Collection](#) that has operator += add the other [Lister](#) or object to the back of its own collection.

The documentation for this class was generated from the following file:

- [lister.h](#)

6.679 f_localizedseparatorlink_class::localizedseparatorlink_type Type Reference

Collaboration diagram for `f_localizedseparatorlink_class::localizedseparatorlink_type`:



Public Member Functions

- final [localizedseparatorlink_class_delete](#)

Private Attributes

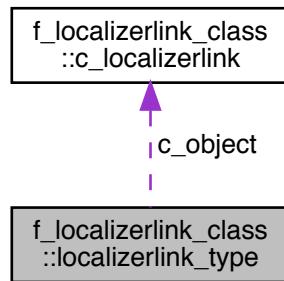
- type([c_localizedseparatorlink](#)), private `c_object`

The documentation for this type was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.680 f_localizerlink_class::localizerlink_type Type Reference

Collaboration diagram for f_localizerlink_class::localizerlink_type:



Public Member Functions

- final **localizerlink_class_delete**

Private Attributes

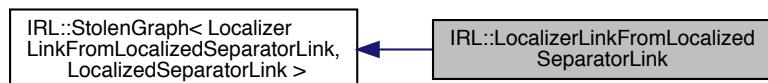
- type([c_localizerlink](#)), private **c_object**

The documentation for this type was generated from the following file:

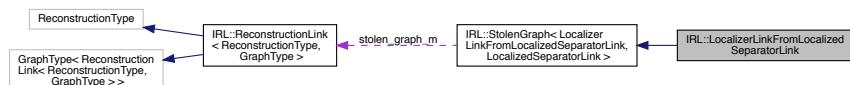
- [f_localizerlink_class.f90](#)

6.681 IRL::LocalizerLinkFromLocalizedSeparatorLink Class Reference

Inheritance diagram for IRL::LocalizerLinkFromLocalizedSeparatorLink:



Collaboration diagram for IRL::LocalizerLinkFromLocalizedSeparatorLink:



Public Member Functions

- [`LocalizerLinkFromLocalizedSeparatorLink`](#) (`void`)=`delete`
Default constructor.
- [`LocalizerLinkFromLocalizedSeparatorLink`](#) (`const LocalizedSeparatorLink *a_localized_separator_link`)
Construct with pointer to `LocalizedSeparatorLink`.
- [`LocalizerLinkFromLocalizedSeparatorLink getNeighbor`](#) (`const UnsignedIndex_t a_neighbor_index`) `const`
- `const PlanarLocalizer & getCurrentReconstruction` (`void`) `const`
- `const LocalizedSeparatorLink * getLinkingReconstructionAddress` (`void`) `const`
Return address of the thing that is linked.
- [`~LocalizerLinkFromLocalizedSeparatorLink`](#) (`void`)=`default`
Default destructor.

Static Public Member Functions

- `static constexpr NullReconstruction getNextReconstruction` (`void`)

The documentation for this class was generated from the following file:

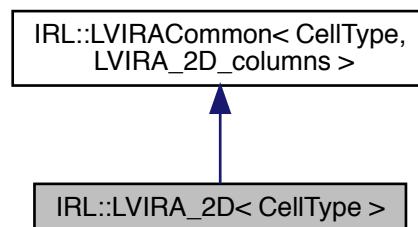
- `localizer_link_from_localized_separator_link.h`

6.682 IRL::LVIRA_2D< CellType > Class Template Reference

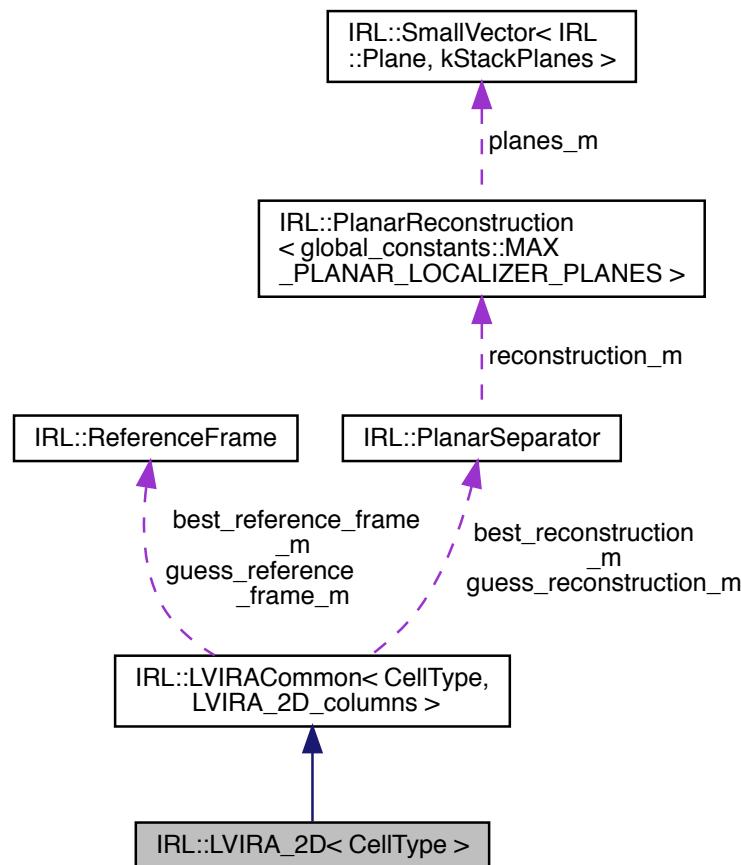
LVIRA class for reconstructions in 2D (x-y plane).

```
#include <lvira_optimization.h>
```

Inheritance diagram for IRL::LVIRA_2D< CellType >:



Collaboration diagram for IRL::LVIRA_2D< CellType >:



Public Types

- using **cell_type** = CellType

Public Member Functions

- **LVIRA_2D** (void)=default
Default constructor.
- **~LVIRA_2D** (void)=default
Default destructor.
- **PlanarSeparator solve** (const **LVIRANeighborhood**< CellType > &a_neighborhood, const **PlanarSeparator** &a_reconstruction)
Perform optimization and return planar separator.
- **void setup** (const **PlanarSeparator** &a_reconstruction)
Initialize simulation parameters that will be needed during the optimization.
- **void updateGuess** (const **Eigen::Matrix< double, columns_m, 1 >** *const a_delta)
Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.

Static Public Attributes

- static constexpr UnsignedIndex_t `columns_m` = `LVIRA_2D_columns`
Number of columns for LVIRA_2D optimization.

Private Member Functions

- `UnitQuaternion getDeltaRotationQuat (const ReferenceFrame &a_reference_frame, const Eigen::Matrix<double, columns_m, 1 > &a_delta)`
Return rotation for LVIRA dictated by elements in a_delta.

Additional Inherited Members

6.682.1 Detailed Description

```
template<class CellType>
class IRL::LVIRA_2D< CellType >
```

LVIRA class for reconstructions in 2D (x-y plane).

6.682.2 Member Function Documentation

6.682.2.1 getDeltaRotationQuat()

```
template<class CellType >
UnitQuaternion IRL::LVIRA_2D< CellType >::getDeltaRotationQuat (
    const ReferenceFrame & a_reference_frame,
    const Eigen::Matrix< double, columns_m, 1 > & a_delta ) [private]
```

Return rotation for LVIRA dictated by elements in a_delta.

The rotation order is:

- Rotate by `a_delta(0)` radians around `a_reference_frame[0]`

The documentation for this class was generated from the following file:

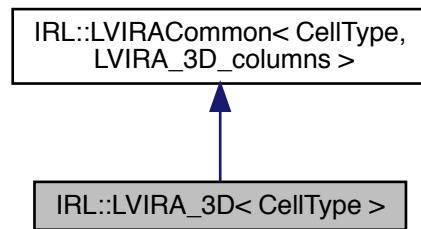
- [lvira_optimization.h](#)

6.683 IRL::LVIRA_3D< CellType > Class Template Reference

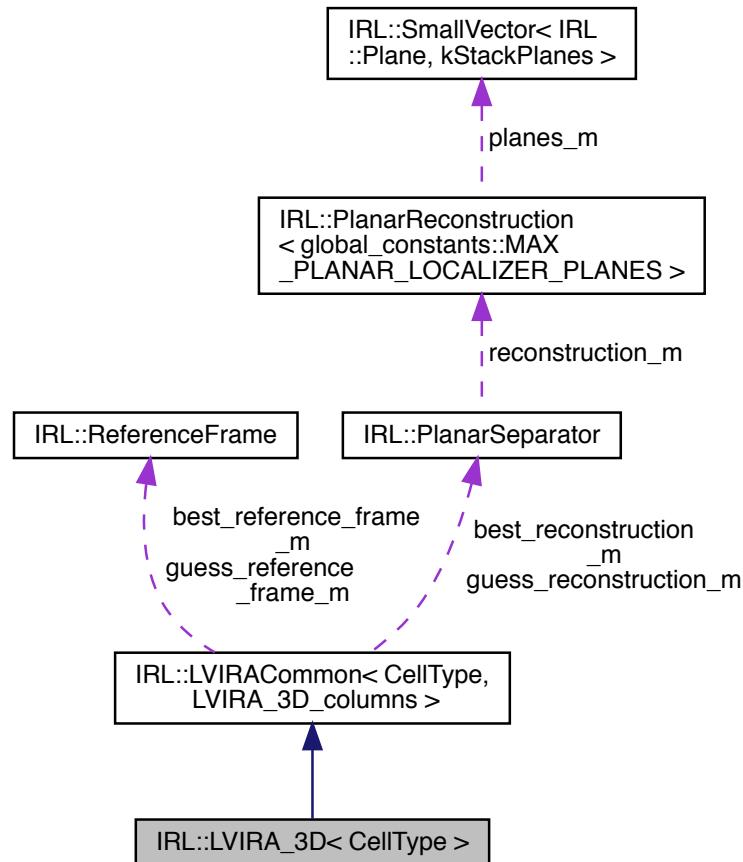
LVIRA class for reconstructions in 3 dimensions.

```
#include <lvira_optimization.h>
```

Inheritance diagram for IRL::LVIRA_3D< CellType >:



Collaboration diagram for IRL::LVIRA_3D< CellType >:



Public Types

- using **cell_type** = CellType

Public Member Functions

- **LVIRA_3D** (void)=default
Default constructor.
- **~LVIRA_3D** (void)=default
Default destructor.
- **PlanarSeparator solve** (const **LVIRANeighborhood**< CellType > &a_neighborhood, const **PlanarSeparator** &a_reconstruction)
Perform optimization and return planar separator.
- **void setup** (const **PlanarSeparator** &a_reconstruction)
Initialize simulation parameters that will be needed during the optimization.
- **void updateGuess** (const **Eigen::Matrix< double, columns_m, 1 >** *const a_delta)
Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.

Static Public Attributes

- static constexpr UnsignedIndex_t **columns_m** = LVIRA_3D_columns
Number of columns for LVIRA_3D optimization.

Private Member Functions

- **UnitQuaternion getDeltaRotationQuat** (const **ReferenceFrame** &a_reference_frame, const Eigen::Matrix<double, **columns_m**, 1 > &a_delta)
Return rotation for LVIRA dictated by elements in a_delta.

Additional Inherited Members

6.683.1 Detailed Description

```
template<class CellType>
class IRL::LVIRA_3D< CellType >
```

LVIRA class for reconstructions in 3 dimensions.

6.683.2 Member Function Documentation

6.683.2.1 getDeltaRotationQuat()

```
template<class CellType >
UnitQuaternion IRL::LVIRA_3D< CellType >::getDeltaRotationQuat (
    const ReferenceFrame & a_reference_frame,
    const Eigen::Matrix< double, columns_m, 1 > & a_delta ) [private]
```

Return rotation for LVIRA dictated by elements in a_delta.

The rotation order is:

- Rotate by a_delta(0) radians around a_reference_frame[0]
- Rotate by a_delta(1) radians around a_reference_frame[1]

The documentation for this class was generated from the following file:

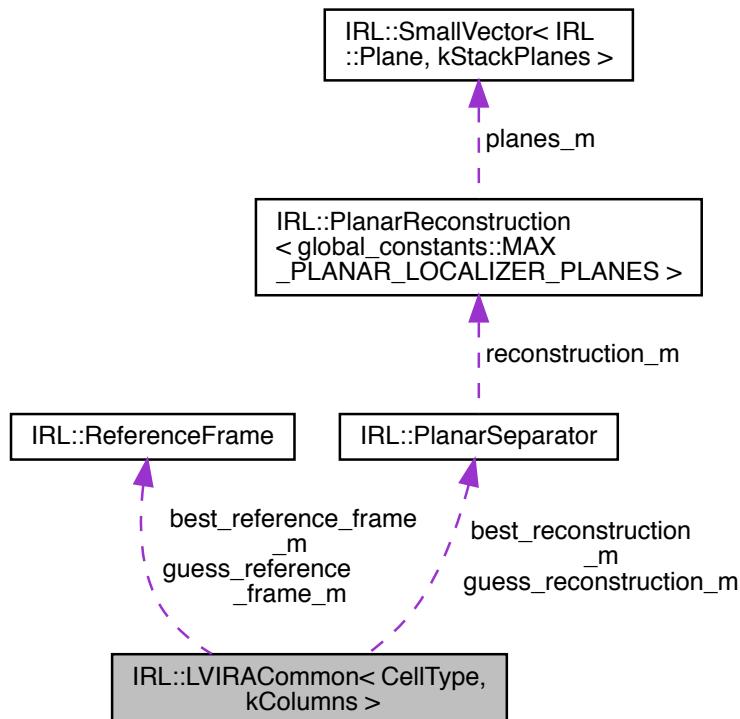
- [lvira_optimization.h](#)

6.684 IRL::LVIRACommon< CellType, kColumns > Class Template Reference

Class to contain data and methods that will be used in all of the specific LVIRA optimization classes.

```
#include <lvira_optimization.h>
```

Collaboration diagram for IRL::LVIRACommon< CellType, kColumns >:



Public Member Functions

- template<class LVIRAType>
`PlanarSeparator runOptimization (LVIRAType *a_ptr_to_LVIRA_object, const LVIRANeighborhood< CellType > &a_neighborhood_geometry, const PlanarSeparator &a_reconstruction)`
Class that takes a pointer to a LVIRA object and executes the optimization, returning the found optimal reconstruction.
- `PlanarSeparator getFinalReconstruction (void)`
Return the final reconstruction to be used.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > calculateVectorError (void)`
Calculate the vector error correct_values_m - guess_values_m where both vectors already have weight applied.
- `bool errorTooHigh (const double a_error)`
Return a boolean stating whether the value is too high or not.
- `bool iterationTooHigh (const UnsignedIndex_t a_iteration)`
Return a boolean stating whether maximum iterations has been exceeded.
- `bool minimumReached (const Eigen::Matrix< double, static_cast< int >(kColumns), 1 > &delta)`

- `Returns whether the minimum is reached.`
- `void increaseLambda (double *a_lambda)`
Increase lambda to decrease step size.
- `void decreaseLambda (double *a_lambda)`
Decrease lambda to increase step size.
- `bool shouldComputeJacobian (const UnsignedIndex_t a_iteration, const UnsignedIndex_t a_last_jacobian)`
Return whether Jacobian should be computed, allowing delayed re-evaluation of the Jacobian.
- `Eigen::Matrix< double, static_cast< int >kColumns, 1 > getJacobianStepSize (void) const`
Return vector of steps in parameters to take when computing Jacobian.
- `double calculateScalarError (void)`
Calculate the error ||correct_values_m - guess_values_m||^2 where both vectors have already had weighting applied.
- `void updateBestGuess (void)`
Save current guess as the best guess.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > calculateChangeInGuess (void)`
Calculate and return vector of differences between guess_values_m and best_values_m. Used in Jacobian calculation.
- `const PlanarSeparator & getBestReconstruction (void)`
Return the best reconstruction found during the optimization procedure.
- `const PlanarSeparator & getGuessReconstruction (void)`
Return a const reference to the guess reconstruction.
- `const ReferenceFrame & getBestReferenceFrame (void)`
Return best guess reference frame.
- `const ReferenceFrame & getGuessReferenceFrame (void)`
Return the guess reference frame.
- `void allocateMatrices (const UnsignedIndex_t a_neighborhood_size)`
Allocate the dynamically sized matrices.
- `void fillGeometryAndWeightVectors (void)`
Set up system and vectors with correct values/weights.
- `void setWeightedGeometryVectorFromReconstruction (const PlanarSeparator &a_reconstruction)`
Update guess_values_m given a reconstruction.
- `PlanarSeparator getReconstructionFromLVIRAParam (const ReferenceFrame &a_reference_frame)`
Return the corresponding planar reconstruction for a given reference frame.

Public Attributes

- `const LVIRANeighborhood< CellType > * neighborhood_m`
Pointer to neighborhood being used in reconstruction.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > weights_m`
Weights to be applied to the correct and guess values.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > correct_values_m`
Weighted vector of correct values we are trying to match.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > guess_values_m`
Weighted vector of guess values that we are trying to match to correct_values_m.
- `PlanarSeparator guess_reconstruction_m`
Guess reconstruction that is used to obtain guess_values_m.
- `ReferenceFrame guess_reference_frame_m`
Guess reference frame used when obtaining guess_reconstruction_m.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > best_values_m`
Weighted vector that resulted in lowest error.
- `PlanarSeparator best_reconstruction_m`
Reconstruction that has resulted in lowest error.
- `ReferenceFrame best_reference_frame_m`
Reference frame associated with the best reconstruction.

Static Public Attributes

- static constexpr double `acceptable_error_m` = $1.0e-4 * 1.0e-4$
If this->`calculateScalarError()` is less than this, exit.
- static constexpr UnsignedIndex_t `maximum_iterations_m` = 20
Maximum number of attempted iterations before exiting.
- static constexpr double `minimum_angle_change_m` = 0.0001745329
Minimum change in angle related delta below which minimum is deemed reached.
- static constexpr double `lambda_increase_m` = 5.0
Increase factor for lambda if more damping needed.
- static constexpr double `lambda_decrease_m` = 1.0 / 10.0
Decrease factor for lambda if new best solution is found.
- static constexpr UnsignedIndex_t `delay_jacobian_amount_m` = 0
Number of iterations to allow between calculating a new Jacobian.
- static constexpr double `fininite_difference_angle_m`
Angle change to use when calculating finite-difference Jacobian.

Private Attributes

- friend `LVIRA_2D< CellType >`
- friend `LVIRA_3D< CellType >`

Friends

- template<class LVIRATypeForDebug >
`class LVIRADebug`

6.684.1 Detailed Description

```
template<class CellType, UnsignedIndex_t kColumns>
class IRL::LVIRACommon< CellType, kColumns >
```

Class to contain data and methods that will be used in all of the specific LVIRA optimization classes.

This class holds common methods and members that are used in the specific LVIRA optimization classes (2D or 3D). This includes variables and members that determine exit criterion for the optimization through Levenberg-Marquardt.

Template parameters:

- `CellType` : Type of cell involved in the neighborhood
- `kColumns` : Number of columns in Jacobian matrix. Equal to number of parameters to optimize for with Levenberg-Marquardt.

6.684.2 Member Function Documentation

6.684.2.1 runOptimization()

```
template<class CellType, UnsignedIndex_t kColumns>
template<class LVIRAType >
PlanarSeparator IRL::LVIRACommon< CellType, kColumns >::runOptimization (
    LVIRAType * a_ptr_to_LVIRA_object,
    const LVIRANeighborhood< CellType > & a_neighborhood_geometry,
    const PlanarSeparator & a_reconstruction )
```

Class that takes a pointer to a LVIRA object and executes the optimization, returning the found optimal reconstruction.

Requirements for LVIRAType:

- Must be either [LVIRA_2D](#) or [LVIRA_3D](#).

Parameters

in	<i>a_ptr_to_LVIRA_object</i>	Pointer to LVIRA object that will be run through optimization.
in	<i>a_neighborhood_geometry</i>	Neighborhood geometry that will be used in the optimization to calculate error we are trying to minimize.
in	<i>a_reconstruction</i>	Initial reconstruction to start LVIRA optimization routine from.

6.684.3 Member Data Documentation

6.684.3.1 fininite_difference_angle_m

```
template<class CellType, UnsignedIndex_t kColumns>
constexpr double IRL::LVIRACommon< CellType, kColumns >::fininite_difference_angle_m [static]
```

Initial value:

```
=  
0.001 * 0.0174533
```

Angle change to use when calculating finite-difference Jacobian.

The documentation for this class was generated from the following file:

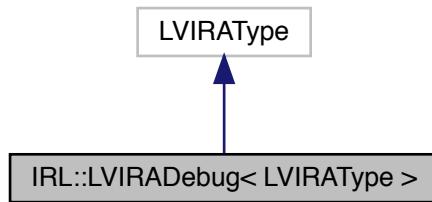
- [lvira_optimization.h](#)

6.685 IRL::LVIRADebug< LVIRAType > Class Template Reference

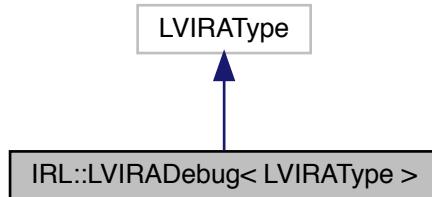
This class just calls the LVIRAType functions but allows debug statements to be printed. The solution path is also saved to be exported and visualized.

```
#include <lvira_optimization.h>
```

Inheritance diagram for IRL::LVIRADebug< LVIRAType >:



Collaboration diagram for IRL::LVIRADebug< LVIRAType >:



Public Member Functions

- [LVIRADebug](#) (void)=default
Default constructor.
- void [updateGuess](#) (const Eigen::Matrix< double, LVIRAType::columns_m, 1 > *const a_delta)
Calls updateGuess from the base class and stores the current reconstruction.
- [PlanarSeparator solve](#) (const [LVIRANeighborhood](#)< typename LVIRAType::cell_type > &a_neighborhood, const [PlanarSeparator](#) &a_reconstruction)
Solve call for debugging LVIRA.
- void [updateBestGuess](#) (void)
Calls updateBestGuess from the base class and stores the current best reconstruction.
- [PlanarSeparator getFinalReconstruction](#) (void)
Shadowed getFinalReconstruction call that writes out stored best reconstructions and then returns the reconstruction.

- void [writeOutPlane](#) (const [PlanarSeparator](#) &a_reconstruction, const std::string &a_prefix, const std::size_t a_iteration_number)
Write the Polygons in the reconstruction out to std::cout, tagged with the given iteration number.
- void [writeOutVolumesAndWeights](#) (void)
Write out the volume fractions and weights to enable visualization of what optimization is driving towards.
- ~[LVIRADebug](#) (void)=default
Default destructor.

Private Attributes

- std::vector< [PlanarSeparator](#) > [guess_reconstruction_history](#)
Saved guess reconstructions encountered during optimization. Note: This will also have reconstructions caused by the calculation of the Jacobian.
- std::vector< [PlanarSeparator](#) > [best_reconstruction_history](#)
Saved best reconstructions accepted during optimization.

6.685.1 Detailed Description

```
template<class LVIRAType>
class IRL::LVIRADebug< LVIRAType >
```

This class just calls the LVIRAType functions but allows debug statements to be printed. The solution path is also saved to be exported and visualized.

The documentation for this class was generated from the following file:

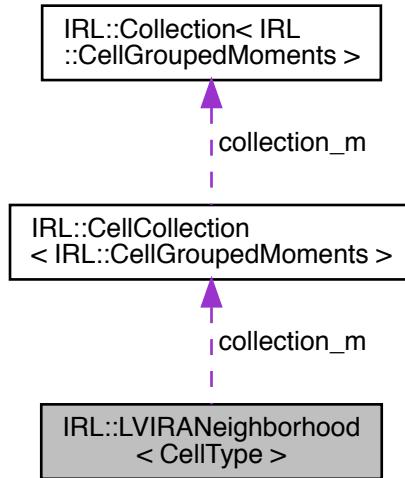
- [lvira_optimization.h](#)

6.686 IRL::LVIRANeighborhood< CellType > Class Template Reference

Neighborhood storage used in the LVIRA optimization routines. This stores the [CellGroupedMoments](#) of the cell and the volume fraction.

```
#include <lvira_neighborhood.h>
```

Collaboration diagram for IRL::LVIRANeighborhood< CellType >:



Public Types

- using **cell_type** = CellType

Public Member Functions

- **LVIRANeighborhood** (void)
Default constructor.
- void **addMember** (const CellType *a_cell, const double *a_volume_fraction)
Construct CellGroupedMoments and add to end of collection.
- void **setMember** (const UnsignedIndex_t a_index, const CellType *a_cell, const double *a_volume_fraction)
Construct CellGroupedMoments and place into collection.
- void **emptyNeighborhood** (void)
Reset neighborhood size to 0.
- void **resize** (const UnsignedIndex_t a_size)
Set size of the neighborhood.
- void **setCenterOfStencil** (const UnsignedIndex_t a_index)
Set the index for the center cell in the collection.
- UnsignedIndex_t **getCenterOfStencilIndex** (void) const
Return the index for the center stencil.
- const CellType & **getCenterCell** (void) const
Return the center cell.
- const double & **getCenterCellStoredMoments** (void) const
Return the center cell moments.
- const CGD::cell_type & **getCell** (const UnsignedIndex_t a_index) const
Return the cell stored at the index.
- const double & **getStoredMoments** (const UnsignedIndex_t a_index) const

Return moments stored at the index.

- `UnsignedIndex_t size (void) const`
Get size of the collection.
- `iterator begin (void) noexcept`
- `const_iterator begin (void) const noexcept`
- `const_iterator end (void) const noexcept`
- `const_iterator cbegin (void) const noexcept`
- `iterator end (void) noexcept`
- `const_iterator cend (void) const noexcept`
- `~LVIRANeighborhood (void)=default`

Default destructor.

Private Types

- `using CGD = CellGroupedMoments< CellType, double >`
- `using iterator = typename CellCollection< CGD >::iterator`
- `using const_iterator = typename CellCollection< CGD >::const_iterator`

Private Member Functions

- `void checkIndex (UnsignedIndex_t a_index) const`
Make sure index is not larger than current collection size.
- `void checkCenterStencilSet (void) const`

Private Attributes

- `CellCollection< CGD > collection_m`
Collection of cells and cell moments.
- `UnsignedIndex_t center_cell_index_m`
Center stencil cell index in the list of added cells.

6.686.1 Detailed Description

```
template<class CellType>
class IRL::LVIRANeighborhood< CellType >
```

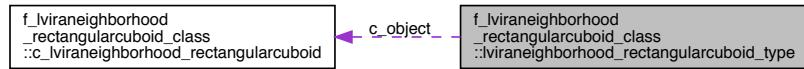
Neighborhood storage used in the LVIRA optimization routines. This stores the `CellGroupedMoments` of the cell and the volume fraction.

The documentation for this class was generated from the following file:

- `lvira_neighborhood.h`

6.687 f_lviraneighborhood_rectangularcuboid_class::lviraneighborhood_rectangularcuboid_type Type Reference

Collaboration diagram for f_lviraneighborhood_rectangularcuboid_class::lviraneighborhood_rectangularcuboid_type:



Public Member Functions

- final lviraneighborhood_rectangularcuboid_class_delete

Private Attributes

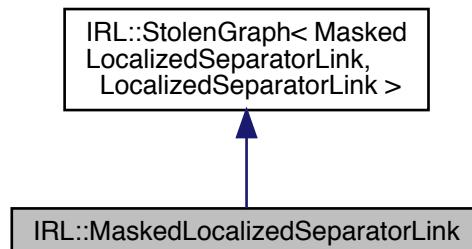
- type([c_lviraneighborhood_rectangularcuboid](#)), private **c_object**

The documentation for this type was generated from the following file:

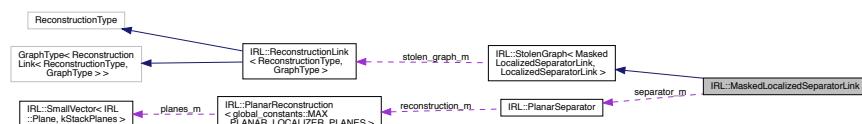
- [f_lviraneighborhood_rectangularcuboid_class.f90](#)

6.688 IRL::MaskedLocalizedSeparatorLink Class Reference

Inheritance diagram for IRL::MaskedLocalizedSeparatorLink:



Collaboration diagram for IRL::MaskedLocalizedSeparatorLink:



Public Member Functions

- `MaskedLocalizedSeparatorLink` (const `LocalizedSeparatorLink` *`a_localized_separator_link`, const `PlanarSeparator` *`a_separator`)
Construct with pointer to LocalizedSeparatorLink.
- `MaskedLocalizedSeparatorLink getNeighbor` (const `UnsignedIndex_t` `a_neighbor_index`) const
- const `PlanarLocalizer` & `getCurrentReconstruction` (void) const
- const `PlanarSeparator` & `getNextReconstruction` (void) const
- const `LocalizedSeparatorLink` * `getLinkingReconstructionAddress` (void) const
Return address of the thing that is linked.
- `~MaskedLocalizedSeparatorLink` (void)=default
Default destructor.

Private Attributes

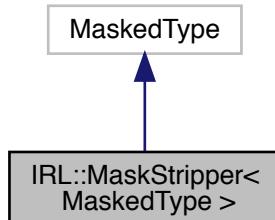
- const `PlanarSeparator` * `separator_m`

The documentation for this class was generated from the following file:

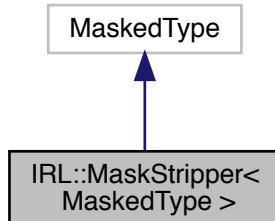
- `masked_localized_separator_link.h`

6.689 IRL::MaskStripper< MaskedType > Class Template Reference

Inheritance diagram for IRL::MaskStripper< MaskedType >:



Collaboration diagram for IRL::MaskStripper< MaskedType >:



Public Types

- using **pt_type** = typename MaskedType::pt_type

Public Member Functions

- pt_type & **operator[]** (const UnsignedIndex_t a_index)
- const pt_type & **operator[]** (const UnsignedIndex_t a_index) const

The documentation for this class was generated from the following file:

- delayed_expandable_vertex_access.h

6.690 IRL::MathVector< kNumberOfElements > Class Template Reference

Public Types

- using **iterator** = typename std::array< double, kNumberOfElements >::iterator
- using **const_iterator** = typename std::array< double, kNumberOfElements >::const_iterator

Public Member Functions

- constexpr **MathVector** (void)=default
Default constructor.
- constexpr **MathVector** (const double a_element_0, const double a_element_1, const double a_element_2)
Constructor for vector given 3 different values.
- double & **operator[]** (const UnsignedIndex_t a_d)
Return address of a_d index of the MathVector.
- const double & **operator[]** (const UnsignedIndex_t a_d) const
Const version of operator[].
- **MathVector operator-** (void) const
- **MathVector & operator+=** (const **MathVector** &a_other_vector)
- **MathVector & operator-=** (const **MathVector** &a_other_vector)
- **MathVector & operator+=** (const double a_scalar)
- **MathVector & operator-=** (const double a_scalar)
- **MathVector & operator*=** (const double a_scalar)
- **MathVector & operator/=** (const double a_scalar)
- UnsignedIndex_t **calculateIndexOfLargestMagnitude** (void) const
Return the index of the component with maximum magnitude.
- double **calculateMagnitude** (void) const
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept
- LargeOffsetIndex_t **getSerializedSize** (void) const
Return size of the serialized normal class in bytes.
- void **serialize** (**ByteBuffer** *a_buffer) const
Serialize the normal and store in the ByteBuffer.
- void **unpackSerialized** (**ByteBuffer** *a_buffer)
Unpack the serialized normal and store.

Static Public Member Functions

- static **MathVector** **fromRawDoublePointer** (const double *a_vec)
- static **MathVector** **fromScalarConstant** (const double a_constant)
- static constexpr UnsignedIndex_t **size** (void)

Length of point vector.

Private Member Functions

- **MathVector** (const double *a_vec)
Constructor for normal given an array containing 3 values.
- **MathVector** (const double a_constant)

Private Attributes

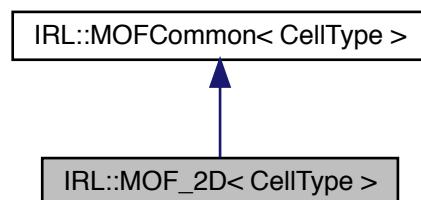
- std::array< double, kNumberOfElements > **elements_m**

The documentation for this class was generated from the following file:

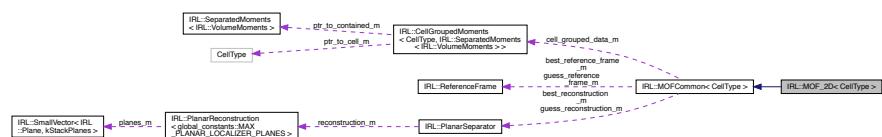
- math_vector.h

6.691 IRL::MOF_2D< CellType > Class Template Reference

Inheritance diagram for IRL::MOF_2D< CellType >:



Collaboration diagram for IRL::MOF_2D< CellType >:



Public Types

- using `cell_type` = CellType

Public Member Functions

- `MOF_2D` (void)=default
Default constructor.
- void `setup` (const `CellGroupedMoments< CellType, SeparatedMoments< VolumeMoments >>` &`a_cell_grouped_data`)
- void `setup` (const `CellGroupedMoments< CellType, SeparatedMoments< VolumeMoments >>` &`a_cell_grouped_data`, const double `a_liquid_weight`, const double `a_gas_weight`)
- `PlanarSeparator solve` (const `CellGroupedMoments< CellType, SeparatedMoments< VolumeMoments >>` &`a_cell_grouped_moments`, const double `a_internal_weight`, const double `a_external_weight`)
Setup and solve the system, returning the found `PlanarSeparator`.
- auto `getDefaultValue` (void) -> const Eigen::Matrix< double, `columns_m`, 1 > &
- void `updateGuess` (Eigen::Matrix< double, `columns_m`, 1 > *`a_delta`)
- bool `minimumReached` (const Eigen::Matrix< double, `columns_m`, 1 > `a_delta`)
- `~MOF_2D` (void)=default
Default destructor.

Static Public Attributes

- static constexpr int `rows_m` = 6
Number of rows for `MOF_2D` optimization.
- static constexpr int `columns_m` = 1
Number of columns for `MOF_2D` optimization.

Private Member Functions

- `UnitQuaternion getDeltaRotationQuat` (const `ReferenceFrame` &`a_reference_frame`, const Eigen::Matrix< double, `columns_m`, 1 > &`a_delta`)

Private Attributes

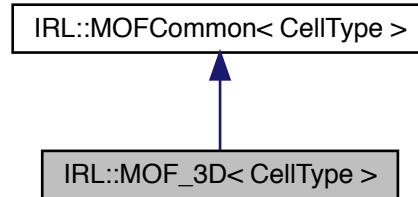
- Eigen::Matrix< double, `columns_m`, 1 > `initial_delta_m`

The documentation for this class was generated from the following file:

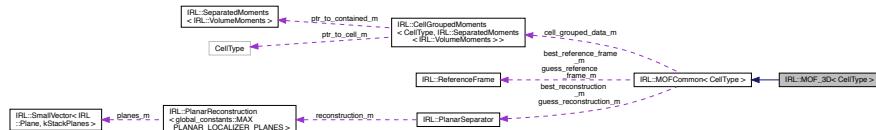
- mof.h

6.692 IRL::MOF_3D< CellType > Class Template Reference

Inheritance diagram for IRL::MOF_3D< CellType >:



Collaboration diagram for IRL::MOF_3D< CellType >:



Public Types

- using **cell_type** = CellType

Public Member Functions

- MOF_3D** (void)=default
Default constructor.
- void **setup** (const **CellGroupedMoments**< CellType, **SeparatedMoments**< VolumeMoments > > &a_cell_< grouped_data)
- void **setup** (const **CellGroupedMoments**< CellType, **SeparatedMoments**< VolumeMoments > > &a_cell_< grouped_data, const double a_liquid_weight, const double a_gas_weight)
- PlanarSeparator** **solve** (const **CellGroupedMoments**< CellType, **SeparatedMoments**< VolumeMoments > > &a_cell_grouped_moments, const double a_internal_weight, const double a_external_weight)

Setup and solve the system, returning the found PlanarSeparator.

- auto **getDefaultInitialDelta** (void) -> const Eigen::Matrix< double, **columns_m**, 1 > &
- void **updateGuess** (Eigen::Matrix< double, **columns_m**, 1 > *a_delta)
- bool **minimumReached** (const Eigen::Matrix< double, **columns_m**, 1 > a_delta)
- ~**MOF_3D** (void)=default

Default destructor.

Static Public Attributes

- static constexpr int `rows_m` = 6
Number of rows for MOF_2D optimization.
- static constexpr int `columns_m` = 2
Number of columns for MOF_2D optimization.

Private Member Functions

- `UnitQuaternion getDeltaRotationQuat (const ReferenceFrame &a_reference_frame, const Eigen::Matrix<double, columns_m, 1 > &a_delta)`

Private Attributes

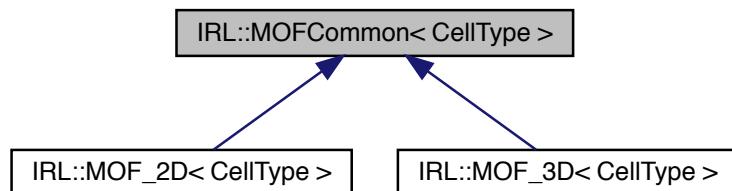
- `Eigen::Matrix< double, columns_m, 1 > initial_delta_m`

The documentation for this class was generated from the following file:

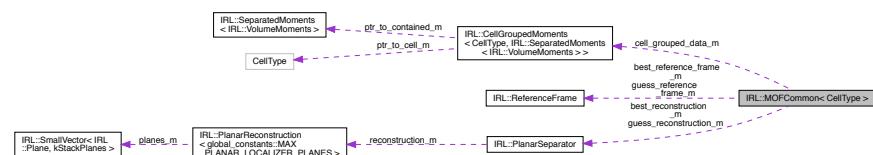
- `mof.h`

6.693 IRL::MOFCommon< CellType > Class Template Reference

Inheritance diagram for IRL::MOFCommon< CellType >:



Collaboration diagram for IRL::MOFCommon< CellType >:



Public Member Functions

- **MOFCommon** (void)=default
Default constructor.
- template<class MOFType >
PlanarSeparator **runOptimization** (MOFType *a_ptr_to_MOF_object, const **CellGroupedMoments**< CellType, **SeparatedMoments**< **VolumeMoments** >> &a_cell_grouped_moments, const double a_internal_weight, const double a_external_weight)
PlanarSeparator **getFinalReconstruction** (void)
*Return the final reconstruction to be used, which includes re-rotation and flipping of the reconstruction (while **getBestReconstruction()** and **getGuessReconstruction()** does not.*
- **double calculateScalarError** (void)
Calculate the error ||a_correct_vector - a_attempt_vector||^2 where both vectors are weighted geometry vectors.
- **Eigen::Matrix< double, 6, 1 > calculateVectorError** (void)
Calculate the vector error a_correct_vector - a_attempt_vector where both vectors are weighted geometry vectors.
- **Eigen::Matrix< double, 6, 1 > calculateChangeInGuess** (void)
Calculate and return vector of differences between guess_values_m and best_values_m. Used in Jacobian calculation.
- **void increaseLambda** (double *a_lambda)
Increase lambda to decrease step size.
- **void decreaseLambda** (double *a_lambda)
Decrease lambda to increase step size.
- **bool errorTooHigh** (const double a_error)
Return a boolean stating whether the value is too high or not.
- **bool iterationTooHigh** (const UnsignedIndex_t a_iteration)
Return a boolean stating whether maximum iterations has been exceeded.
- **bool shouldComputeJacobian** (const UnsignedIndex_t a_iteration, const UnsignedIndex_t a_last_jacobian)
Return whether Jacobian should be computed, allowing delayed re-evaluation of the Jacobian.
- **void updateBestGuess** (void)
Save current guess as the best guess.
- **~MOFCommon** (void)=default
Default destructor.

Static Public Attributes

- static constexpr double **acceptable_error_m** = 1.0e-4 * 1.0e-4
*If this->**calculateScalarError()** is less than this, exit.*
- static constexpr UnsignedIndex_t **maximum_iterations_m** = 40
Maximum number of attempted iterations before exiting.
- static constexpr double **minimum_angle_change_m** = 0.0001 * 0.0174533
Minimum change in angle related delta below which minimum is deemed reached.
- static constexpr double **lambda_increase_m** = 5.0
Increase factor for lambda if more damping needed.
- static constexpr double **lambda_decrease_m** = 1.0 / 10.0
Decrease factor for lambda if new best solution is found.
- static constexpr UnsignedIndex_t **delay_jacobian_amount_m** = 0
Number of iterations to allow between calculating a new Jacobian.
- static constexpr double **initial_angle_m**
Initial angle to use when first calculating Jacobian, equal to 5 degrees in radians.

Private Member Functions

- void `fillGeometryAndWeightVectors` (double `a_liquid_weight`, double `a_gas_weight`)

Setup the weight and correct vectors.
- void `setWeightedGeometryVectorFromReconstruction` (const `PlanarSeparator &a_reconstruction`)

Perform geometric integration for volume moments, apply weighting, and save to `guess_values_m`.
- `PlanarSeparator getReconstructionFromParam` (const `ReferenceFrame &a_reference_frame`)

Given a reference frame, return the `PlanarSeparator` that matches the volume fraction with the normal from the reference frame.
- const `PlanarSeparator & getBestReconstruction` (void)

Return the best reconstruction found during the optimization procedure.
- const `PlanarSeparator & getGuessReconstruction` (void)

Return a const reference to the guess reconstruction.
- double `getInternalVolume` (void)

Return the internal (under plane) volume from the `cell_grouped_data_m`.
- const `Pt & getInternalCentroid` (void)

Return the internal (under plane) centroid from the `cell_grouped_data_m`.
- const `Pt & getExternalCentroid` (void)

Return the external (above plane) centroid from the `cell_grouped_data_m`.

Private Attributes

- friend `MOF_2D< CellType >`
- friend `MOF_3D< CellType >`
- const `CellGroupedMoments< CellType, SeparatedMoments< VolumeMoments > > * cell_grouped_data_m`
- double `geom_scale_factor_m`

Scale factor used for geometry in order to give dimension independence. Equal to $\text{vol}^{(1/3)}$.
- Eigen::Matrix< double, 6, 1 > `weights_m`

Weights to be applied to the corect and guess `SeparatedMoments<VolumeMoments>/Surface Area`.
- Eigen::Matrix< double, 6, 1 > `correct_values_m`

Weighted vector of correct values we are trying to match.
- double `volume_fraction_m`

Volume fraction in cell being reconstructed.
- Eigen::Matrix< double, 6, 1 > `guess_values_m`

Weighted vector of guess values that we are trying to match to `correct_values_m`.
- `PlanarSeparator guess_reconstruction_m`

Guess reconstruction that is used to obtain `guess_values_m`.
- `ReferenceFrame guess_reference_frame_m`

Guess reference frame used when obtaining `guess_reconstruction_m`.
- Eigen::Matrix< double, 6, 1 > `best_values_m`

Weighted vector that resulted in lowest error.
- `PlanarSeparator best_reconstruction_m`

Reconstruction that has resulted in lowest error.
- `ReferenceFrame best_reference_frame_m`

Reference frame associated with the best reconstruction.

6.693.1 Member Data Documentation

6.693.1.1 initial_angle_m

```
template<class CellType >
constexpr double IRL::MOFCommon< CellType >::initial_angle_m [static]
```

Initial value:

```
= 0.0001 * 0.0174533
```

Initial angle to use when first calculating Jacobian, equal to 5 degrees in radians.

The documentation for this class was generated from the following file:

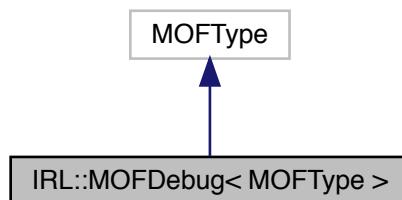
- mof.h

6.694 IRL::MOFDebug< MOFType > Class Template Reference

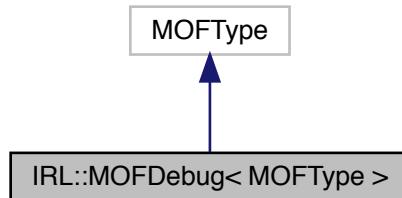
This class just calls the MOFType functions but allows debug statements to be printed.

```
#include <mof.h>
```

Inheritance diagram for IRL::MOFDebug< MOFType >:



Collaboration diagram for IRL::MOFDebug< MOFType >:



Public Member Functions

- [MOFDebug \(void\)=default](#)
Default constructor.
- [PlanarSeparator solve \(const CellGroupedMoments< typename MOFType::cell_type, SeparatedMoments< VolumeMoments >> &a_cell_grouped_moments, const double a_internal_weight, const double a_external_weight\)](#)
Setup and solve the system, returning the found PlanarSeparator.
- [void updateGuess \(Eigen::Matrix< double, MOFType::columns_m, 1 > *a_delta\)](#)
Calls updateGuess from the base class and stores the current reconstruction.
- [void updateBestGuess \(void\)](#)
Calls updateBestGuess from the base class and stores the current reconstruction.
- [PlanarSeparator getFinalReconstruction \(void\)](#)
Shadowed getFinalReconstruction call that writes out stored best reconstructions and then returns the reconstruction.
- [~MOFDebug \(void\)=default](#)
Default destructor.

Private Member Functions

- [void writeOutPlane \(const PlanarSeparator &a_reconstruction, const std::string &a_prefix, const std::size_t a_iteration_number\)](#)
Write the ConvexPolygons in the reconstruction out to std::cout, tagged with the given iteration number.
- [void writeOutCentroidsAndWeights \(void\)](#)
Write out the centroids and weights to enable visualization of what optimization is driving towards.

Private Attributes

- [std::vector< PlanarSeparator > guess_reconstruction_history](#)
Saved guess reconstructions encountered during optimization.
- [std::vector< PlanarSeparator > best_reconstruction_history](#)
Saved best reconstructions accepted during optimization.

6.694.1 Detailed Description

```
template<class MOFType>
class IRL::MOFDebug< MOFType >
```

This class just calls the MOFType functions but allows debug statements to be printed.

This class masks MOFType functions to allow calling the function and then all printing any debug information wanted. It also stores the guessed reconstructions and saved "best" reconstructions in order to reprint them out later. The printing format of this is to plot the resulting vertices of the plane polygons to be plotted by R2P/references/r2p_history_plot.m. To do this, the information printed to screen needs to be copied to R2P/references/reconstruction_history.m.

Requirements for MOFType class:

- MOFType class must meet all requirements needed by the [LevenbergMarquardt](#) class in [optimizers.h](#). Right now, this is mainly [MOF_2D](#) and [MOF_3d](#).

The documentation for this class was generated from the following file:

- mof.h

6.695 IRL::MomentCalculationType< VertexType, ProxyType, Enable > Class Template Reference

The documentation for this class was generated from the following file:

- segmented_decomposed_polytope.h

6.696 IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polygon< ProxyType >::value > > Class Template Reference

Inheritance diagram for IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polygon< ProxyType >::value > >:



Collaboration diagram for IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polygon< ProxyType >::value > >:



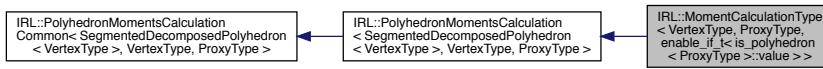
Additional Inherited Members

The documentation for this class was generated from the following file:

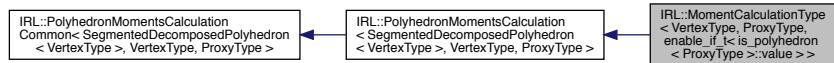
- segmented_decomposed_polytope.h

6.697 IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polyhedron< ProxyType >::value > > Class Template Reference

Inheritance diagram for IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polyhedron< ProxyType >::value > >:



Collaboration diagram for IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polyhedron< ProxyType >::value >>:



Additional Inherited Members

The documentation for this class was generated from the following file:

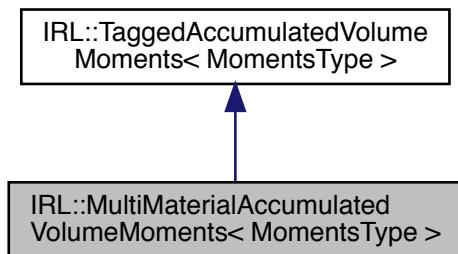
- segmented_decomposed_polytope.h

6.698 IRL::MultiMaterialAccumulatedVolumeMoments< MomentsType > Class Template Reference

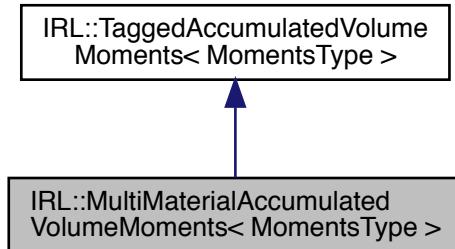
This class is a specialization of [TaggedAccumulatedVolumeMoments<MomentsType>](#) that allows the computation of SeparatedMoments<MomentsType> for the target ID and the encompassing volume. Encompassing volume can be either given, or computed as the moments of all other phases.

```
#include <multimaterial_accumulated_volume_moments.h>
```

Inheritance diagram for IRL::MultiMaterialAccumulatedVolumeMoments< MomentsType >:



Collaboration diagram for IRL::MultiMaterialAccumulatedVolumeMoments< MomentsType >:



Public Member Functions

- [SeparatedMoments< MomentsType > calculateSeparatedMoments \(const UnsignedIndex_t a_id\)](#)
- template<class GeometryType> [SeparatedMoments< MomentsType > calculateSeparatedMoments \(const UnsignedIndex_t a_id, const GeometryType &a_encompassing_geometry\)](#)

Additional Inherited Members

6.698.1 Detailed Description

```
template<class MomentsType>
class IRL::MultiMaterialAccumulatedVolumeMoments< MomentsType >
```

This class is a specialization of [TaggedAccumulatedVolumeMoments<MomentsType>](#) that allows the computation of SeparatedMoments<MomentsType> for the target ID and the encompassing volume. Encompassing volume can be either given, or computed as the moments of all other phases.

The documentation for this class was generated from the following file:

- multimaterial_accumulated_volume_moments.h

6.699 f_tagged_accumvm_sepvm_class::multiplybyvolume Interface Reference

Public Member Functions

- subroutine [tagged_accumvm_sepvm_class_multiplybyvolume \(this\)](#)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.700 f_tagged_accumvm_vm_class::multiplybyvolume Interface Reference

Public Member Functions

- subroutine **tagged_accumvm_vm_class_multiplybyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.701 f_vman_class::multiplybyvolume Interface Reference

Public Member Functions

- subroutine **vman_class_multiplybyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.702 f_sepvm_class::multiplybyvolume Interface Reference

Public Member Functions

- subroutine **sepvm_class_multiplybyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.703 f_sepvm_doubles3_class::multiplybyvolume Interface Reference

Public Member Functions

- subroutine **sepvm_doubles3_class_multiplybyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.704 f_objectallocationserver_localizedseparatorlink_class::new Interface Reference

Public Member Functions

- subroutine **objectallocationserver_localizedseparatorlink_class_new** (this, a_number_to_allocate)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_localizedseparatorlink_class.f90](#)

6.705 f_elviraneighborhood_class::new Interface Reference

Public Member Functions

- subroutine **elviraneighborhood_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_elviraneighborhood_class.f90](#)

6.706 f_objectallocationserver_planarlocalizer_class::new Interface Reference

Public Member Functions

- subroutine **objectallocationserver_planarlocalizer_class_new** (this, a_number_to_allocate)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_planarlocalizer_class.f90](#)

6.707 f_planarlocalizer_class::new Interface Reference

Public Member Functions

- subroutine **planarlocalizer_class_new** (this)
- subroutine **planarlocalizer_class_newfromobjectallocationserver** (this, a_object_allocation_server)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.708 f_tet_class::new Interface Reference

Public Member Functions

- subroutine **tet_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_tet_class.f90](#)

6.709 f_polygon_class::new Interface Reference

Public Member Functions

- subroutine **polygon_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.710 f_dodecahedron_class::new Interface Reference

Public Member Functions

- subroutine **dodecahedron_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_dodecahedron_class.f90](#)

6.711 f_planarseparator_class::new Interface Reference

Public Member Functions

- subroutine **planarseparator_class_new** (this)
- subroutine **planarseparator_class_newfromobjectallocationserver** (this, a_object_allocation_server)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.712 f_rectangularcuboid_class::new Interface Reference

Public Member Functions

- subroutine **rectangularcuboid_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.713 f_vman_class::new Interface Reference

Public Member Functions

- subroutine **vman_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.714 f_tri_class::new Interface Reference

Public Member Functions

- subroutine **tri_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.715 f_cappeddodecahedron_doubles3_class::new Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_doubles3_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.716 f_tagged_accumvm_vm_class::new Interface Reference

Public Member Functions

- subroutine **tagged_accumvm_vm_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_vm_class.f90](#)

6.717 f_tagged_accumlistedvm_vman_class::new Interface Reference

Public Member Functions

- subroutine **tagged_accumlistedvm_vman_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.718 f_dividedpolygon_class::new Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.719 f_polyhedron24_class::new Interface Reference

Public Member Functions

- subroutine **polyhedron24_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.720 f_polyhedron24_doubles3_class::new Interface Reference

Public Member Functions

- subroutine **polyhedron24_doubles3_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.721 f_listedvm_vman_class::new Interface Reference

Public Member Functions

- subroutine **listedvm_vman_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_listedvm_vman_class.f90](#)

6.722 f_bytebuffer_class::new Interface Reference

Public Member Functions

- subroutine **bytebuffer_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.723 f_objectallocationserver_planarseparator_class::new Interface Reference

Public Member Functions

- subroutine **objectallocationserver_planarseparator_class_new** (this, a_number_to_allocate)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_planarseparator_class.f90](#)

6.724 f_r2pneighborhood_rectangularcuboid_class::new Interface Reference

Public Member Functions

- subroutine **r2pneighborhood_rectangularcuboid_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.725 f_localizedseparatorlink_class::new Interface Reference

Public Member Functions

- subroutine **localizedseparatorlink_class_new** (this, a_planar_localizer, a_planar_separator)
- subroutine **localizedseparatorlink_class_newfromobjectallocationserver** (this, a_object_allocation_server, a_planar_localizer, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.726 f_localizerlink_class::new Interface Reference

Public Member Functions

- subroutine **localizerlink_class_new** (this, a_planar_localizer)
- subroutine **localizerlink_class_newfromobjectallocationserver** (this, a_object_allocation_server, a_planar_localizer)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.727 f_objectallocationserver_localizerlink_class::new Interface Reference

Public Member Functions

- subroutine **objectallocationserver_localizerlink_class_new** (this, a_number_to_allocate)

The documentation for this interface was generated from the following file:

- [f_objectallocationserver_localizerlink_class.f90](#)

6.728 f_sepvm_class::new Interface Reference

Public Member Functions

- subroutine **sepvm_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.729 f_sepvm_doubles3_class::new Interface Reference

Public Member Functions

- subroutine **sepvm_doubles3_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.730 f_lviraneighborhood_rectangularcuboid_class::new Interface Reference

Public Member Functions

- subroutine **lviraneighborhood_rectangularcuboid_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_lviraneighborhood_rectangularcuboid_class.f90](#)

6.731 f_tagged_accumvm_sepvm_class::new Interface Reference

Public Member Functions

- subroutine **tagged_accumvm_sepvm_class_new** (this)

The documentation for this interface was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.732 f_cappeddodecahedron_class::new Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_class_new** (this)

The documentation for this interface was generated from the following file:

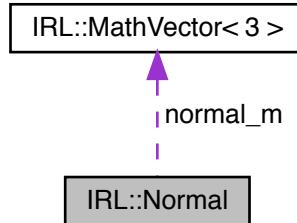
- [f_cappeddodecahedron_class.f90](#)

6.733 IRL::Normal Class Reference

A normal vector in 3D space.

```
#include <normal.h>
```

Collaboration diagram for IRL::Normal:



Public Member Functions

- **constexpr Normal** (void)

Default constructor with zero normal.
- **constexpr Normal** (const double a_normal_x, const double a_normal_y, const double a_normal_z)

Constructor for normal given 3 different values.
- **double & operator[]** (const UnsignedIndex_t a_d)

Return address of a_d index of normal.
- **const double & operator[]** (const UnsignedIndex_t a_d) const

Const version of operator[].
- **Normal operator-** (void) const

Overload unary - operator.
- **Normal & operator+=** (const Normal &a_rhs)

Increment by another normal.
- **Normal & operator/=** (const double a_value)

Overload /= operator to divide all components by scalar.

- `Normal & operator*=(const double a_value)`
*Overload *= operator to multiply all components by scalar.*
- `Normal & operator=(const double a_value)`
Assign all elements in Normal the a value.
- `bool operator==(const Normal &a_normal) const`
Comparison operator to return if two normals are the same.
- `bool operator!=(const Normal &a_normal) const`
Comparison operator for const to return if two normals are not the same.
- `UnsignedIndex_t maxMagnitudeComponent(void) const`
Return the index of the component with maximum magnitude.
- `double calculateMagnitude(void) const`
Const version of calculateMagnitude.
- `void normalize(void)`
Normalize the normal.
- `iterator begin(void) noexcept`
- `const_iterator begin(void) const noexcept`
- `const_iterator cbegin(void) const noexcept`
- `iterator end(void) noexcept`
- `const_iterator end(void) const noexcept`
- `const_iterator cend(void) const noexcept`
- `LargeOffsetIndex_t getSerializedSize(void) const`
Return size of the serialized normal class in bytes.
- `void serialize(ByteBuffer *a_buffer) const`
Serialize the normal and store in the ByteBuffer.
- `void unpackSerialized(ByteBuffer *a_buffer)`
Unpack the serialized normal and store.
- `~Normal(void)=default`
Default destructor.

Static Public Member Functions

- `static Normal normalized(const double a_normal_x, const double a_normal_y, const double a_normal_z)`
- `static Normal fromRawDoublePointer(const double *a_normal)`
- `static Normal fromRawDoublePointerNormalized(const double *a_normal)`
- `static Normal fromPt(const Pt &a_pt)`
- `static Normal fromPtNormalized(const Pt &a_pt)`
- `static Pt toPt(const Normal &a_normal)`
- `static Normal fromScalarConstant(const double a_constant)`
- `static constexpr UnsignedIndex_t size(void)`
Length of point vector.

Private Types

- `using iterator = Vec3::iterator`
- `using const_iterator = Vec3::const_iterator`

Private Member Functions

- `constexpr Normal (const double *a_normal)`
Constructor for normal given an array containing 3 values.
- `Normal (const Pt &a_pt)`
Set a pt as a normal and normalize. Usually occurs from subtraction of two points.
- `Normal (const double a_constant)`

Private Attributes

- `Vec3 normal_m`
N_x, N_y, N_z values of normal.

6.733.1 Detailed Description

A normal vector in 3D space.

The documentation for this class was generated from the following file:

- `normal.h`

6.734 f_tagged_accumvm_sepvm_class::normalizebyvolume Interface Reference

Public Member Functions

- subroutine `tagged_accumvm_sepvm_class_normalizebyvolume (this)`

The documentation for this interface was generated from the following file:

- `f_tagged_accumvm_sepvm_class.f90`

6.735 f_tagged_accumvm_vm_class::normalizebyvolume Interface Reference

Public Member Functions

- subroutine `tagged_accumvm_vm_class_normalizebyvolume (this)`

The documentation for this interface was generated from the following file:

- `f_tagged_accumvm_vm_class.f90`

6.736 f_sepvm_class::normalizebyvolume Interface Reference

Public Member Functions

- subroutine **sepvm_class_normalizebyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_class.f90](#)

6.737 f_sepvm_doubles3_class::normalizebyvolume Interface Reference

Public Member Functions

- subroutine **sepvm_doubles3_class_normalizebyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.738 f_vman_class::normalizebyvolume Interface Reference

Public Member Functions

- subroutine **vman_class_normalizebyvolume** (this)

The documentation for this interface was generated from the following file:

- [f_vman_class.f90](#)

6.739 IRL::NullReconstruction Class Reference

```
#include <null_reconstruction.h>
```

6.739.1 Detailed Description

This is simply a blank class that is passed when successive reconstructions are done (such as in a Localized Separator) to signify that there are no more reconstructions to be done. For instance, a [PlanarSeparator](#) does not have any more reconstructions after it (its `getNextReconstruction(void)` method should return nothing). Instead, it returns this [NullReconstruction](#), which can have a function specialized for it where the moments of the passed geometry object are simply calculated and returned.

The documentation for this class was generated from the following file:

- [null_reconstruction.h](#)

6.740 IRL::ObjectAllocationServer< ObjectType > Class Template Reference

Public Member Functions

- `ObjectAllocationServer (const LargeOffsetIndex_t a_number_to_allocate)`
- `ObjectType * getNewObject (void)`
- `ObjectAllocationServer (const ObjectAllocationServer &other)=delete`
- `ObjectAllocationServer (ObjectAllocationServer &&other) noexcept=delete`
- `ObjectAllocationServer & operator= (const ObjectAllocationServer &other)=delete`
- `ObjectAllocationServer & operator= (ObjectAllocationServer &&other) noexcept=delete`

Private Member Functions

- `ObjectType * serveNextAvailableObject (void)`
- `LargeOffsetIndex_t getIndexForNextObjectToServe (void)`
- `void incrementNumberOfServedObjects (void)`
- `LargeOffsetIndex_t indexOfNextObjectToServe (void) const`

Private Attributes

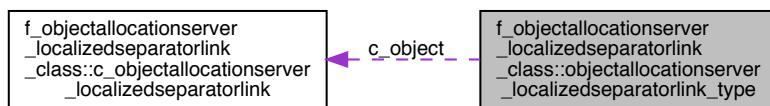
- `BlockObjectAllocation< ObjectType > object_allocation_m`
- `LargeOffsetIndex_t number_of_objects_served_m`

The documentation for this class was generated from the following file:

- `object_allocation_server.h`

6.741 f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_type Type Reference

Collaboration diagram for `f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_type`:



Public Member Functions

- `final objectallocationserver_localizedseparatorlink_class_delete`

Private Attributes

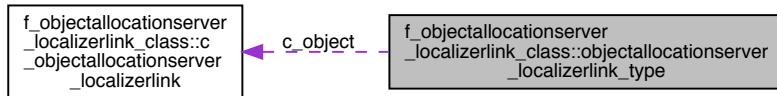
- type(`c_objectallocationserver_localizedseparatorlink`), private `c_object`

The documentation for this type was generated from the following file:

- `f_objectallocationserver_localizedseparatorlink_class.f90`

6.742 `f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_type` Type Reference

Collaboration diagram for `f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_type`:



Public Member Functions

- final `objectallocationserver_localizerlink_class_delete`

Private Attributes

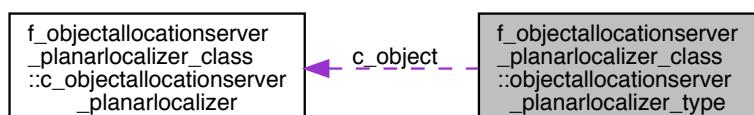
- type(`c_objectallocationserver_localizerlink`), private `c_object`

The documentation for this type was generated from the following file:

- `f_objectallocationserver_localizerlink_class.f90`

6.743 `f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer_type` Type Reference

Collaboration diagram for `f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer_type`:



Public Member Functions

- final **objectallocationserver_planarlocalizer_class_delete**

Private Attributes

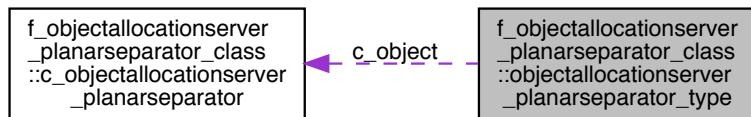
- type([c_objectallocationserver_planarlocalizer](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_objectallocationserver_planarlocalizer_class.f90](#)

6.744 **f_objectallocationserver_planarseparator_class::objectallocationserver_planarseparator_type** Type Reference

Collaboration diagram for **f_objectallocationserver_planarseparator_class::objectallocationserver_planarsePARATOR_type**:



Public Member Functions

- final **objectallocationserver_planarseparator_class_delete**

Private Attributes

- type([c_objectallocationserver_planarseparator](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_objectallocationserver_planarseparator_class.f90](#)

6.745 **IRL::PartitionByNormal< MomentsContainerType >** Class Template Reference

This is a class that takes a list of objects and separates the list into two partitions, which are represented by the sums of the objects.

```
#include <partition_by_normal_vector.h>
```

Public Member Functions

- [PartitionByNormal \(void\)](#)
Default constructor.
- [PartitionByNormal \(const MomentsContainerType *a_polygon_container_ptr\)](#)
Constructor that takes pointer to list of polygons to be partitioned.
- [void setup \(const MomentsContainerType *a_polygon_container_ptr\)](#)
Set pointer, then setup the system before passing to [KMeans](#).
- [void setup \(void\)](#)
Setup the system before passing to [KMeans](#).
- [std::array< typename MomentsContainerType::contained_type, 2 > getPartitionedObjects \(void\)](#)
Write partitioned objects to a provided buffer.
- [~PartitionByNormal \(void\)=default](#)
Default constructor.

Private Types

- [using iterator = typename MomentsContainerType::iterator](#)
- [using const_iterator = typename MomentsContainerType::const_iterator](#)

Private Member Functions

- [bool isDone \(void\)](#)
Check if partitioning isdone by seeing if the found normals did not change during iteration.
- [bool iterationTooHigh \(const UnsignedIndex_t a_iteration_number\)](#)
Return bool indicating if max iterations has been surpassed.
- [void setupNextIteration \(void\)](#)
Setup the next iteration by updating old normals to previous iteration value and zeroing partitioned moments.
- [UnsignedIndex_t findCorrectPartition \(const typename MomentsContainerType::contained_type &a_element\)](#)
Find which partition the normal belongs in.
- [void addElementToPartition \(const UnsignedIndex_t a_partition, const typename MomentsContainerType::contained_type &a_element\)](#)
Add the element to the indicated partition.
- [const_iterator begin \(void\) noexcept](#)
- [const_iterator begin \(void\) const noexcept](#)
- [const_iterator end \(void\) const noexcept](#)
- [const_iterator cbegin \(void\) const noexcept](#)
- [const_iterator end \(void\) noexcept](#)
- [const_iterator cend \(void\) const noexcept](#)
- [MomentsContainerType::contained_type getObjectWithMostVolume \(void\)](#)
Find and return the object in the list with the most associated volume.
- [MomentsContainerType::contained_type getObjectMostDifferent \(const Normal &a_normal\)](#)
Find and return the object in the list with the normal most different from a_normal.

Private Attributes

- [friend KMeans](#)
- [const MomentsContainerType * list_ptr_m](#)
- [std::array< typename MomentsContainerType::contained_type, 2 > partitioned_objects_m](#)
- [std::array< Normal, 2 > old_partition_normals_m](#)

Static Private Attributes

- static constexpr UnsignedIndex_t **max_iteration_number** = 40

6.745.1 Detailed Description

```
template<class MomentsContainerType>
class IRL::PartitionByNormal< MomentsContainerType >
```

This is a class that takes a list of objects and separates the list into two partitions, which are represented by the sums of the objects.

This class is used to split a list of objects into two separate groups depending on normal vectors. The normal vectors contained as part of the objects in the list are assumed to not necessarily be normalized, and normalization is done before they are used. Instead of saving the elements in each partition, the sum of the objects (through operator `+=`) is computed for each partition. This is done by providing this class (once setup) to the [KMeans](#) class, where `KMeans::solve()` is then used.

Requirements for ContainerType:

- Const_Iterator that allows traversing the entire initial set held in `list_ptr_m`.
- `const Normal& normal(void)` : A method that returns a [Normal](#) (or const reference to one) to be used for comparison. This [Normal](#) does not need to be normalized, will be normalized during partitioning.
- `Object& overload+=(const Object&)` : A method that overloads `+=` to enable summing of the elements into their partitions.

The documentation for this class was generated from the following file:

- `partition_by_normal_vector.h`

6.746 IRL::PassToNestedType Struct Reference

Static Public Member Functions

- `template<class ReturnType , class ReconstructionType , class SimplexType >
static void handleVolumeCompletelyInternalToPlanarReconstruction (const SimplexType &a_simplex,
const ReconstructionType &a_reconstruction, ReturnType *a_moments_to_return)`

The documentation for this struct was generated from the following file:

- `handle_enclosed_volume.h`

6.747 IRL::PathGraphNode< NodeType > Class Template Reference

Class to inherit from to provide linking between reconstructions, where the linking exists PER RECONSTRUCTION. This also follows a directed path-graph (linear graph), with a starting node and constant direction to the end node. The interface is kept the same as for the [UnDirectedGraphNode](#) in order to facilitate reuse in the cutting routines. All neighbor indices will simply point to the next node in the path-graph.

```
#include <path_graph_node.h>
```

Public Types

- using **node_type** = NodeType

Public Member Functions

- **PathGraphNode** (void)
Default constructor.
- void **setEdgeConnectivity** (const NodeType *a_neighbor_ptr)
Supplied the pointer for the neighbor at the index.
- void **setId** (const UnsignedIndex_t a_id)
Set unique Id for this node in the graph.
- UnsignedIndex_t **getId** (void) const
- bool **isIdSet** (void) const
- bool **hasNeighbor** (const UnsignedIndex_t a_neighbor_index=0) const
- const NodeType & **getNeighbor** (const UnsignedIndex_t a_neighbor_index=0) const
- const NodeType * **getNeighborAddress** (const UnsignedIndex_t a_neighbor_index=0) const
- const NodeType * **getNodeMemoryAddress** (void) const
- **~PathGraphNode** (void)=default
Default destructor.

Private Attributes

- const NodeType * **next_neighbor_m**
- UnsignedIndex_t **id_m**

6.747.1 Detailed Description

```
template<class NodeType>
class IRL::PathGraphNode< NodeType >
```

Class to inherit from to provide linking between reconstructions, where the linking exists PER RECONSTRUCTION. This also follows a directed path-graph (linear graph), with a starting node and constant direction to the end node. The interface is kept the same as for the [UnDirectedGraphNode](#) in order to facilitate reuse in the cutting routines. All neighbor indices will simply point to the next node in the path-graph.

The documentation for this class was generated from the following file:

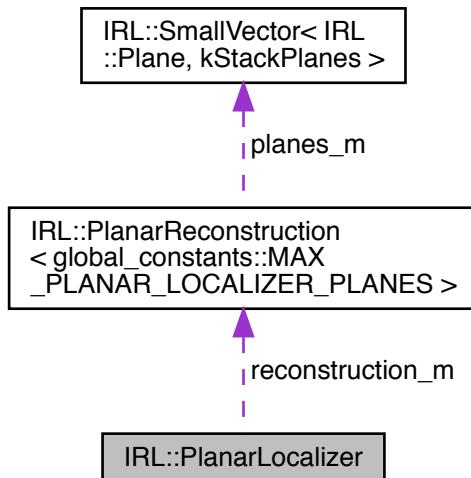
- path_graph_node.h

6.748 IRL::PlanarLocalizer Class Reference

A planar representation of a convex polyhedron to localize integrations.

```
#include <planar_localizer.h>
```

Collaboration diagram for IRL::PlanarLocalizer:



Public Member Functions

- `PlanarLocalizer (void)=default`
Default constructor.
- `PlanarLocalizer (const PlanarLocalizer &a_other)=default`
- `PlanarLocalizer & operator= (const PlanarLocalizer &a_other)=default`
- `UnsignedIndex_t getNumberOfPlanes (void) const`
Return the number of planes used for the reconstruction.
- `void setNumberOfPlanes (const UnsignedIndex_t a_number_of_future_planes)`
Shrink/enlarge vector of planes to given number.
- `void zeroNumberOfPlanes (void)`
Directly set the number of planes to 0.
- `Plane & operator[] (const UnsignedIndex_t a_p)`
Overload [] to access planes_m through reference.
- `const Plane & operator[] (const UnsignedIndex_t a_p) const`
Overload [] to access planes_m through const reference.
- `void addPlane (const Plane &a_plane)`
Add a new plane to the list.
- `void addBeginningPlane (const Plane &a_plane)`
Insert new plane at start of planes.
- `void removePlane (const UnsignedIndex_t a_p)`
Remove the plane given by index a_p

- const [PlanarLocalizer](#) & **getCurrentReconstruction** (void) const
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept
- LargeOffsetIndex_t **getSerializedSize** (void) const
Return size of the serialized PlanarLocalizer.
- void **serialize** (ByteBuffer *a_buffer) const
Serialize and pack the planes.
- void **unpackSerialized** (ByteBuffer *a_buffer)
Unpack the planes.
- **~PlanarLocalizer** (void)=default
Default destructor.

Static Public Member Functions

- static [PlanarLocalizer](#) **fromOnePlane** (const [Plane](#) &a_plane)
- static [PlanarLocalizer](#) **fromTwoPlanes** (const [Plane](#) &a_plane_0, const [Plane](#) &a_plane_1)
- static constexpr double **flip** (void)
Since localizers are always convex, never flip.
- static constexpr bool **isFlipped** (void)
Since localizers are always convex, never flipped.
- static constexpr bool **isNotFlipped** (void)
Return if cutting for gas phase is needed.
- static constexpr [NullReconstruction](#) **getNextReconstruction** (void)

Private Types

- using **PlanarReconstructionBase** = [PlanarReconstruction](#)< global_constants::MAX_PLANAR_LOCALIZERS_ZER_PLANES >
- using **iterator** = [PlanarReconstructionBase](#)::iterator
- using **const_iterator** = [PlanarReconstructionBase](#)::const_iterator

Private Member Functions

- [PlanarLocalizer](#) (const [Plane](#) &a_plane)
Constructor when given a single plane.
- [PlanarLocalizer](#) (const [Plane](#) &a_plane_0, const [Plane](#) &a_plane_1)
Constructor when given two planes.

Private Attributes

- [PlanarReconstructionBase](#) **reconstruction_m**

Friends

- std::ostream & **operator<<** (std::ostream &out, const [PlanarLocalizer](#) &a_reconstruction)

6.748.1 Detailed Description

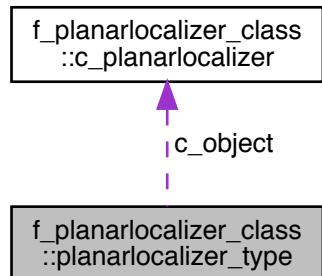
A planar representation of a convex polyhedron to localize integrations.

The documentation for this class was generated from the following file:

- `planar_localizer.h`

6.749 `f_planarlocalizer_class::planarlocalizer_type` Type Reference

Collaboration diagram for `f_planarlocalizer_class::planarlocalizer_type`:



Public Member Functions

- final `planarlocalizer_class_delete`

Private Attributes

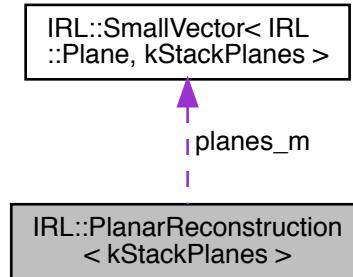
- `type(c_planarlocalizer)`, private `c_object`

The documentation for this type was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.750 IRL::PlanarReconstruction< kStackPlanes > Class Template Reference

Collaboration diagram for IRL::PlanarReconstruction< kStackPlanes >:



Public Member Functions

- **PlanarReconstruction (void)=default**
Default constructor.
- **UnsignedIndex_t getNumberOfPlanes (void) const**
Return the number of planes used for the reconstruction.
- **void reserve (const UnsignedIndex_t a_number_of_future_planes)**
Reserve space for a number of planes that will (presumably) be added later.
- **void setNumberOfPlanes (const UnsignedIndex_t a_number_of_future_planes)**
Shrink/enlarge vector of planes to given number.
- **void zeroNumberOfPlanes (void)**
Directly set the number of planes to 0.
- **Plane & operator[] (const UnsignedIndex_t a_p)**
Overload [] to access planes_m through reference.
- **const Plane & operator[] (const UnsignedIndex_t a_p) const**
Overload [] to access planes_m through const reference.
- **void addPlane (const Plane &a_plane)**
Add a new plane to the list.
- **void addBeginningPlane (const Plane &a_plane)**
Insert new plane at start of planes.
- template<UnsignedIndex_t kStackVectorSize>
void **setDistances (const SmallVector< double, kStackVectorSize > &a_distances)**
Set distances to the planes in the reconstruction.
- **void removePlane (const UnsignedIndex_t a_p)**
Remove the plane given by index a_p
- **LargeOffsetIndex_t getSerializedSize (void) const**
Return size of the serialized PlanarReconstruction.
- **void serialize (ByteBuffer *a_buffer) const**
Serialize and pack the planes.
- **void unpackSerialized (ByteBuffer *a_buffer)**

Unpack the planes and store.

- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept
- **~PlanarReconstruction** (void)=default

Default destructor.

Static Public Member Functions

- static **PlanarReconstruction fromOnePlane** (const **Plane** &a_plane)
- static **PlanarReconstruction fromTwoPlanes** (const **Plane** &a_plane_0, const **Plane** &a_plane_1)

Private Types

- using **iterator** = typename **SmallVector< Plane, kStackPlanes >::iterator**
- using **const_iterator** = typename **SmallVector< Plane, kStackPlanes >::const_iterator**

Private Member Functions

- **PlanarReconstruction** (const **Plane** &a_plane)
Construct with one given plane.
- **PlanarReconstruction** (const **Plane** &a_plane_0, const **Plane** &a_plane_1)
Construct with two given planes.
- void **checkIfStaticAllocationExceeded** (void) const

Private Attributes

- **SmallVector< Plane, kStackPlanes > planes_m**
Planes in reconstruction.

Friends

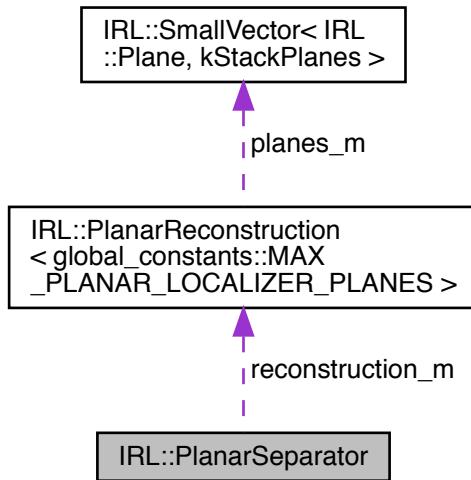
- class **PlanarSeparator**
- class **PlanarLocalizer**

The documentation for this class was generated from the following file:

- planar_reconstruction.h

6.751 IRL::PlanarSeparator Class Reference

Collaboration diagram for IRL::PlanarSeparator:



Public Member Functions

- **PlanarSeparator (void)**
Default constructor, initialize one plane that is far below everything.
- **UnsignedIndex_t getNumberOfPlanes (void) const**
Return the number of planes used for the reconstruction.
- **void setNumberOfPlanes (const UnsignedIndex_t a_number_of_future_planes)**
Shrink/enlarge vector of planes to given number.
- **void zeroNumberOfPlanes (void)**
Directly set the number of planes to 0.
- **Plane & operator[] (const UnsignedIndex_t a_p)**
Overload [] to access planes_m through reference.
- **const Plane & operator[] (const UnsignedIndex_t a_p) const**
Overload [] to access planes_m through const reference.
- **void addPlane (const Plane &a_plane)**
Add a new plane to the reconstruction.
- **template<UnsignedIndex_t kStackVectorSize> void setDistances (const SmallVector< double, kStackVectorSize > &a_distances)**
Set distances for planes by given a pointer to an array of doubles this->[getNumberOfPlanes\(\)](#) long.
- **void removePlane (const UnsignedIndex_t a_p)**
Remove the plane given by index a_p
- **double flip (void) const**
Return value of flip_m
- **void setFlip (const double a_flip_value)**
Set flip directly.

- void `doNotFlipCutting` (void)
Make it so regular cutting is used (liquid phase, below plane, found by cutting).
- void `flipCutting` (void)
Make it so that cutting by this reconstruction will be flipped (gas phase found through cutting).
- bool `isFlipped` (void) const
Return if cutting for gas phase is needed.
- bool `isNotFlipped` (void) const
Return if cutting for gas phase is needed.
- void `zeroPlanes` (void)
Set reconstruction to cause single-phase cell.
- const `PlanarSeparator` & `getCurrentReconstruction` (void) const
- LargeOffsetIndex_t `getSerializedSize` (void) const
Return size of the serialized `PlanarSeparator`.
- void `serialize` (ByteBuffer *a_buffer) const
Serialize and pack the planes and flip_cut_m.
- void `unpackSerialized` (ByteBuffer *a_buffer)
Unpack the planes and flip_cut_m.
- iterator `begin` (void) noexcept
- const_iterator `begin` (void) const noexcept
- const_iterator `cbegin` (void) const noexcept
- iterator `end` (void) noexcept
- const_iterator `end` (void) const noexcept
- const_iterator `cend` (void) const noexcept
- `~PlanarSeparator` (void)=default
Default destructor.

Static Public Member Functions

- static `PlanarSeparator` `fromOnePlane` (const `Plane` &a_plane)
- static `PlanarSeparator` `fromTwoPlanes` (const `Plane` &a_plane_0, const `Plane` &a_plane_1, const double a_flip_indicator)
- static constexpr `NullReconstruction` `getNextReconstruction` (void)

Private Types

- using `PlanarReconstructionBase` = `PlanarReconstruction< global_constants::MAX_PLANAR_LOCALIZEROPLANES >`
- using `iterator` = `PlanarReconstructionBase::iterator`
- using `const_iterator` = `PlanarReconstructionBase::const_iterator`

Private Member Functions

- `PlanarSeparator` (const `Plane` &a_plane)
Constructor when given a single plane.
- `PlanarSeparator` (const `Plane` &a_plane_0, const `Plane` &a_plane_1, const double a_flip_indicator)
Constructor when given two planes.

Private Attributes

- `PlanarReconstructionBase` **reconstruction_m**
- double **flip_cut_m**

Used to flip the phase cut for.

Friends

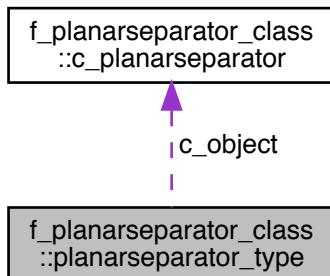
- `std::ostream & operator<< (std::ostream &out, const PlanarSeparator &a_reconstruction)`

The documentation for this class was generated from the following file:

- `planar_separator.h`

6.752 f_planarseparator_class::planarseparator_type Type Reference

Collaboration diagram for `f_planarseparator_class::planarsePARATOR_type`:



Public Member Functions

- final `planarseparator_class_delete`

Private Attributes

- `type(c_planarseparator)`, private **c_object**

The documentation for this type was generated from the following file:

- `f_planarseparator_class.f90`

6.753 IRL::PlanarSeparatorPathGroup Class Reference

This is a class that helps organize PlanarSeparatorPath objects to be coherent in memory and facilitates their control more easily. It will continue to store all PlanarSeparatorPath objects that were added to it, but in most routine only the active ones will be used. Here, active means those that are present in priority_order_by_id_m.

```
#include <planar_separator_path_group.h>
```

Public Member Functions

- void **addPlanarSeparatorPath** (const PlanarSeparatorPath &a_planar_separator_path)
- void **addPlanarSeparatorPath** (PlanarSeparatorPath &&a_planar_separator_path)
- void **addPlanarSeparatorPath** (const PlanarSeparatorPath &a_planar_separator_path, const UnsignedIndex_t a_id)
- void **addPlanarSeparatorPath** (PlanarSeparatorPath &&a_planar_separator_path, const UnsignedIndex_t a_id)
- template<class StorageListType >
void **setPriorityOrder** (const StorageListType &a_priority_order)
StorageListType must have a []() operator and size() method.
- void **setPriorityOrder** (const std::initializer_list< UnsignedIndex_t > &a_priority_order)
- void **setPriorityOrder** (const UnsignedIndex_t a_number_of_entries, UnsignedIndex_t *a_priority_order)
- PlanarSeparatorPath & **getReconstructionByPriority** (const UnsignedIndex_t a_priority_index)
- const PlanarSeparatorPath & **getReconstructionByPriority** (const UnsignedIndex_t a_priority_index) const
- const PlanarSeparatorPath & **getReconstructionById** (const UnsignedIndex_t a_id) const
- PlanarSeparatorPath & **getReconstructionById** (const UnsignedIndex_t a_id)
- const PlanarSeparatorPath & **getFirstReconstruction** (void) const
- const PlanarSeparatorPath & **getCurrentReconstruction** (void) const
- NullReconstruction **getNextReconstruction** (void) const

Private Member Functions

- void **setLinkingByPriorityList** (void)
- bool **isTagKnown** (const UnsignedIndex_t a_tag) const
- bool **isTagNew** (const UnsignedIndex_t a_tag) const
- bool **allPrioritiesExist** (void) const

Private Attributes

- std::vector< UnsignedIndex_t > **priority_order_by_id_m**
- std::unordered_map< UnsignedIndex_t, PlanarSeparatorPath > **id_mapping_m**

6.753.1 Detailed Description

This is a class that helps organize PlanarSeparatorPath objects to be coherent in memory and facilitates their control more easily. It will continue to store all PlanarSeparatorPath objects that were added to it, but in most routine only the active ones will be used. Here, active means those that are present in priority_order_by_id_m.

The documentation for this class was generated from the following file:

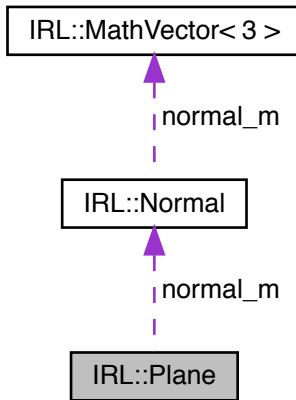
- planar_separator_path_group.h

6.754 IRL::Plane Class Reference

[Plane](#) defined by $\mathbf{n} \cdot \mathbf{x} - d = 0$, where the normal points from liquid to gas.

```
#include <plane.h>
```

Collaboration diagram for IRL::Plane:



Public Member Functions

- [Plane \(void\)](#)
Default constructor.
- [Plane \(const Normal &a_normal, const double a_distance\)](#)
Construct a plane given a [Normal](#) and a distance.
- [Normal & normal \(void\)](#)
Return a copy of the normal forming this plane.
- [const Normal & normal \(void\) const](#)
Return a const reference to the normal forming this plane.
- [double & distance \(void\)](#)
Return value of the distance of the plane.
- [const double & distance \(void\) const](#)
Return const value to the distance of the plane.
- [bool operator==\(const Plane &a_other_plane\) const](#)
Check if two planes are the same.
- [bool operator!=\(const Plane &a_other_plane\) const](#)
Check if two planes are different.
- template<class PtType >
[double signedDistanceToPoint \(const PtType &a_pt\) const](#)
Return signed distance from this plane to the supplied point, where negative is underneath the plane.
- [LargeOffsetIndex_t getSerializedSize \(void\) const](#)
Return size of the serialized point class in bytes.
- [void serialize \(ByteBuffer *a_buffer\) const](#)

- `void unpackSerialized (ByteBuffer *a_buffer)`
Unpack the serialized normal and store.
- `Plane generateFlippedPlane (void) const`
- `~Plane (void)=default`
Default destructor.

Private Member Functions

- `void checkValidNormal (void) const`
Make sure normal have magnitude of 0.0 or 1.0.

Private Attributes

- `Normal normal_m`
Normal for the plane.
- `double distance_m`
Normal distance to the plane.

6.754.1 Detailed Description

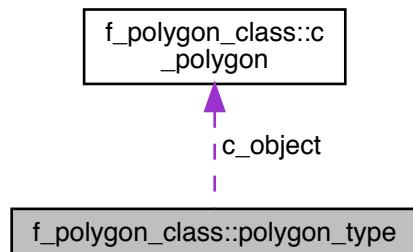
`Plane` defined by $\mathbf{n} \cdot \mathbf{x} - d = 0$, where the normal points from liquid to gas.

The documentation for this class was generated from the following file:

- `plane.h`

6.755 f_polygon_class::polygon_type Type Reference

Collaboration diagram for `f_polygon_class::polygon_type`:



Public Member Functions

- final **polygon_class_delete**

Private Attributes

- type(**c_polygon**), private **c_object**

The documentation for this type was generated from the following file:

- **f_polygon_class.f90**

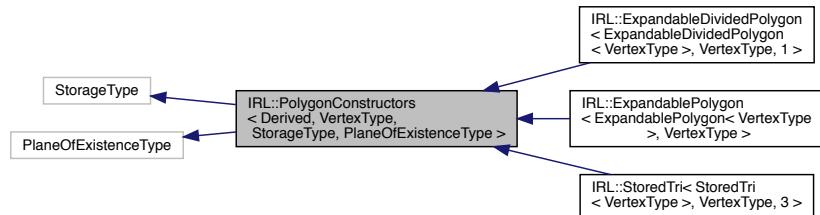
6.756 IRL::PolygonBase< VertexType > Class Template Reference

The documentation for this class was generated from the following file:

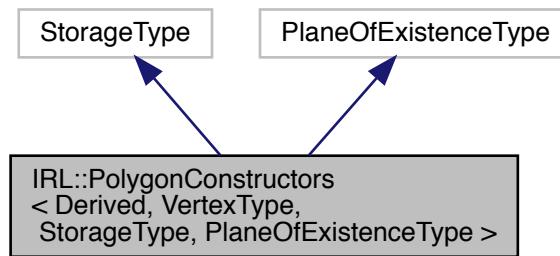
- expandable_pt_list.h

6.757 IRL::PolygonConstructors< Derived, VertexType, StorageType, PlaneOfExistenceType > Class Template Reference

Inheritance diagram for IRL::PolygonConstructors< Derived, VertexType, StorageType, PlaneOfExistenceType >:



Collaboration diagram for IRL::PolygonConstructors< Derived, VertexType, StorageType, PlaneOfExistenceType >:



Public Member Functions

- **PolygonConstructors** (std::initializer_list< VertexType > a_list, const Plane &a_plane_of_existence)

Static Public Member Functions

- static Derived **fromRawPtPointer** (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts)
- static Derived **fromRawDoublePointer** (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs)
- static Derived **fromRawPtPointer** (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts, const Plane &a_plane_of_existence)
- static Derived **fromRawDoublePointer** (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs, const Plane &a_plane_of_existence)

Private Member Functions

- **PolygonConstructors** (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts, const Plane &a_plane_of_existence)

Construct n-pts from array of pts.
- **PolygonConstructors** (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs, const Plane &a_plane_of_existence)

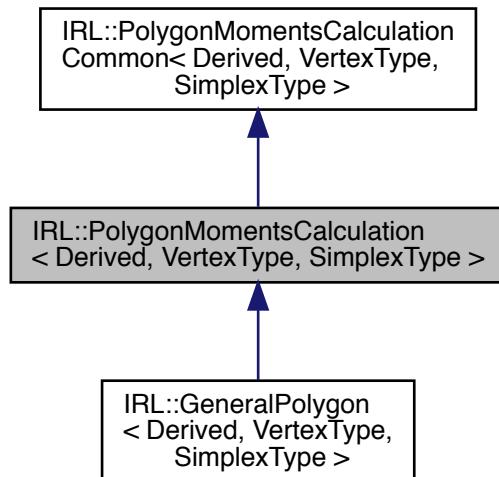
Construct n-pts from array of doubles.

The documentation for this class was generated from the following file:

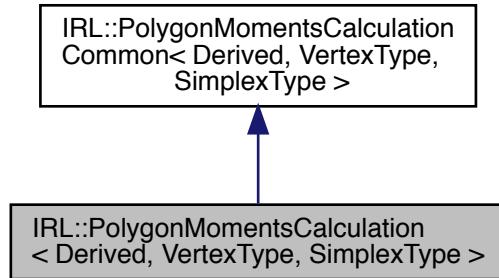
- polygon_storage_types.h

6.758 IRL::PolygonMomentsCalculation< Derived, VertexType, SimplexType > Class Template Reference

Inheritance diagram for IRL::PolygonMomentsCalculation< Derived, VertexType, SimplexType >:



Collaboration diagram for IRL::PolygonMomentsCalculation< Derived, VertexType, SimplexType >:



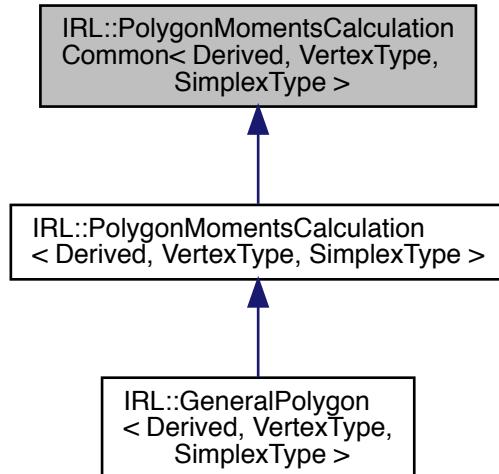
Additional Inherited Members

The documentation for this class was generated from the following file:

- polygon_moments_calculation.h

6.759 IRL::PolygonMomentsCalculationCommon< Derived, VertexType, SimplexType > Class Template Reference

Inheritance diagram for IRL::PolygonMomentsCalculationCommon< Derived, VertexType, SimplexType >:



Public Member Functions

- UnsignedIndex_t **getNumberOfSimplicesInDecomposition** (void) const
- SimplexType **getSimplexFromDecomposition** (const UnsignedIndex_t a_tri_number_to_get) const
- const VertexType & **operator[]** (const UnsignedIndex_t a_index) const
- UnsignedIndex_t **getNumberOfVertices** (void) const
- const Plane & **getPlaneOfExistence** (void) const
- Volume **calculateVolume** (void) const

Calculate and return volume of the tri.
- Volume **calculateAbsoluteVolume** (void) const
- Volume **calculateConvexVolume** (void) const
- double **calculateSign** (void) const
- Pt **calculateCentroid** (void) const

Calculate and return centroid of the tri.
- VolumeMoments **calculateMoments** () const

Calculate and return volume weighted VolumeMoments.
- VolumeMomentsAndNormal **calculateVolumeMomentsAndNormal** () const

Calculate and return volume weighted VolumeMoments.

Private Member Functions

- const Derived & **getDerived** (void) const
- double **calculate2DArea** (const UnsignedIndex_t a_index_0, const UnsignedIndex_t a_index_1) const
- std::array< UnsignedIndex_t, 3 > **getDimensionsOrderedForAscendingFaceNormalMagnitude** (Normal a_normal) const

The documentation for this class was generated from the following file:

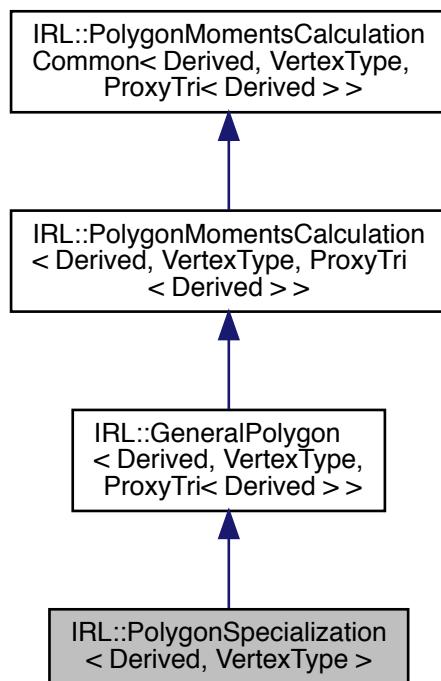
- polygon_moments_calculation.h

6.760 IRL::PolygonSpecialization< Derived, VertexType > Class Template Reference

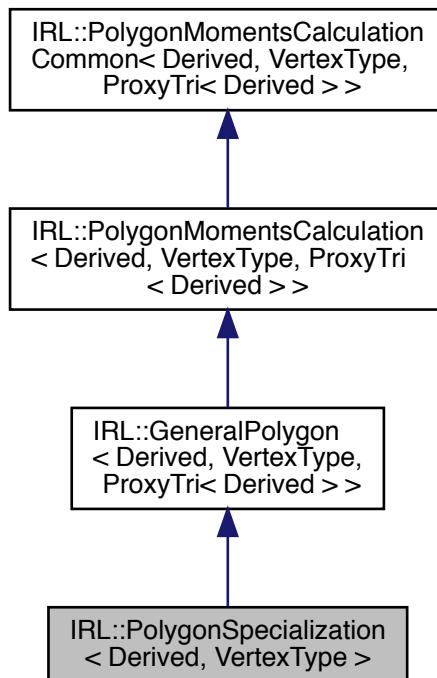
A polygon from a given plane.

```
#include <polygon.h>
```

Inheritance diagram for IRL::PolygonSpecialization< Derived, VertexType >:



Collaboration diagram for IRL::PolygonSpecialization< Derived, VertexType >:



Public Types

- using **pt_type** = VertexType

Public Member Functions

- **HalfEdgePolygon< pt_type > generateHalfEdgeVersion (void) const**
- template<class HalfEdgePolygonType >
void setHalfEdgeVersion (HalfEdgePolygonType *a_half_edge_version) const
- **UnsignedIndex_t getNumberOfSimplicesInDecomposition (void) const**
- **std::array< UnsignedIndex_t, 3 > getSimplexIndicesFromDecomposition (UnsignedIndex_t a_tri_number_to_get) const**
- **ProxyTri< Derived > getSimplexFromDecomposition (const UnsignedIndex_t a_tri_number_to_get) const**
Returns the triangle.

6.760.1 Detailed Description

```
template<class Derived, class VertexType>
class IRL::PolygonSpecialization< Derived, VertexType >
```

A polygon from a given plane.

The documentation for this class was generated from the following file:

- polygon.h

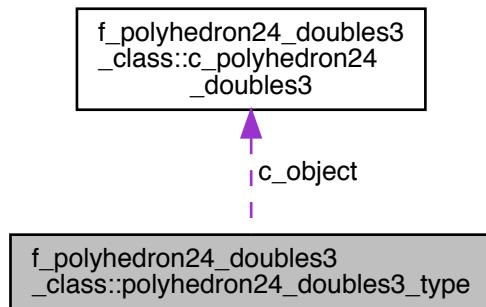
6.761 IRL::Polyhedron< VertexType, kMaxNumberOfPts > Class Template Reference

The documentation for this class was generated from the following file:

- `pt_list.h`

6.762 f_polyhedron24_doubles3_class::polyhedron24_doubles3_type Type Reference

Collaboration diagram for `f_polyhedron24_doubles3_class::polyhedron24_doubles3_type`:



Public Member Functions

- final `polyhedron24_doubles3_class_delete`

Private Attributes

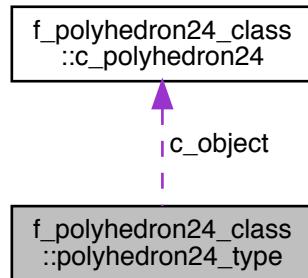
- type(`c_polyhedron24_doubles3`), private `c_object`

The documentation for this type was generated from the following file:

- `f_polyhedron24_doubles3_class.f90`

6.763 f_polyhedron24_class::polyhedron24_type Type Reference

Collaboration diagram for f_polyhedron24_class::polyhedron24_type:



Public Member Functions

- final **polyhedron24_class_delete**

Private Attributes

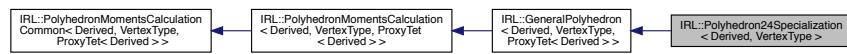
- type([c_polyhedron24](#)), private **c_object**

The documentation for this type was generated from the following file:

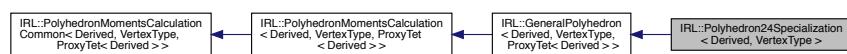
- [f_polyhedron24_class.f90](#)

6.764 IRL::Polyhedron24Specialization< Derived, VertexType > Class Template Reference

Inheritance diagram for IRL::Polyhedron24Specialization< Derived, VertexType >:



Collaboration diagram for IRL::Polyhedron24Specialization< Derived, VertexType >:



Public Member Functions

- `HalfEdgePolyhedron< VertexType > generateHalfEdgeVersion (void) const`
- `template<class HalfEdgePolyhedronType>`
`void setHalfEdgeVersion (HalfEdgePolyhedronType *a_half_edge_version) const`
- `ProxyTet< Derived > getSimplexFromDecomposition (const UnsignedIndex_t a_tet) const`
- `void adjustCapToMatchVolume (const Volume a_correct_volume)`

Static Public Member Functions

- `static constexpr UnsignedIndex_t getNumberOfSimplicesInDecomposition (void)`
- `static constexpr std::array< UnsignedIndex_t, 4 > getSimplexIndicesFromDecomposition (const UnsignedIndex_t a_tet)`

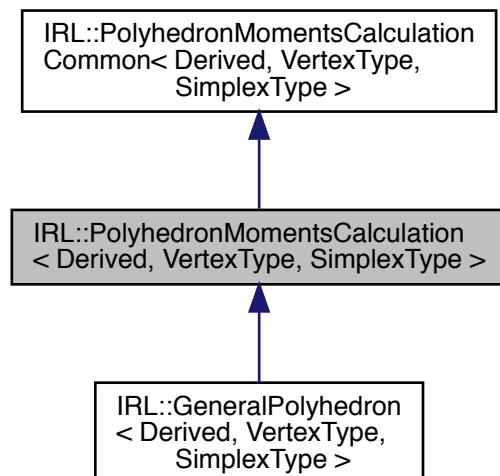
Additional Inherited Members

The documentation for this class was generated from the following file:

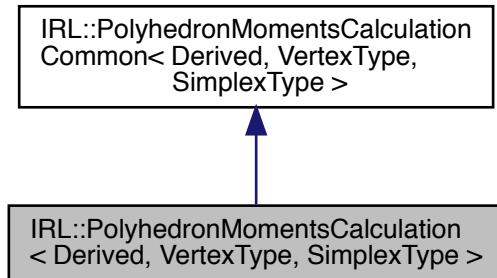
- `polyhedron_24.h`

6.765 IRL::PolyhedronMomentsCalculation< Derived, VertexType, SimplexType > Class Template Reference

Inheritance diagram for IRL::PolyhedronMomentsCalculation< Derived, VertexType, SimplexType >:



Collaboration diagram for IRL::PolyhedronMomentsCalculation< Derived, VertexType, SimplexType >:



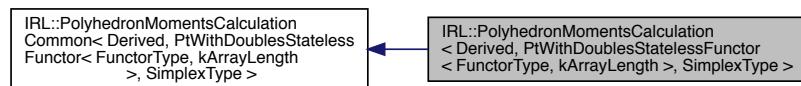
Additional Inherited Members

The documentation for this class was generated from the following file:

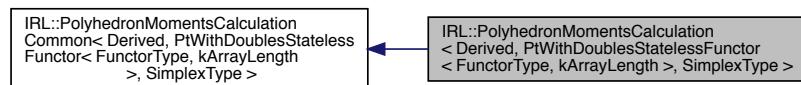
- polyhedron_moments_calculation.h

6.766 IRL::PolyhedronMomentsCalculation< Derived, PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, SimplexType > Class Template Reference

Inheritance diagram for IRL::PolyhedronMomentsCalculation< Derived, PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, SimplexType >:



Collaboration diagram for IRL::PolyhedronMomentsCalculation< Derived, PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, SimplexType >:



Public Member Functions

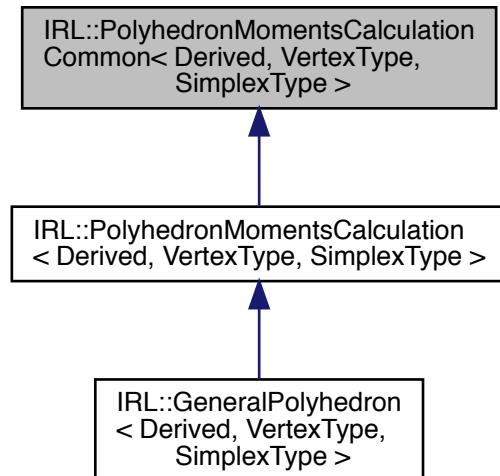
- `VolumeMomentsAndDoubles< kArrayLength > calculateVolumeMomentsAndDoubles (void) const`

The documentation for this class was generated from the following file:

- `polyhedron_moments_calculation.h`

6.767 IRL::PolyhedronMomentsCalculationCommon< Derived, VertexType, SimplexType > Class Template Reference

Inheritance diagram for IRL::PolyhedronMomentsCalculationCommon< Derived, VertexType, SimplexType >:



Public Member Functions

- `UnsignedIndex_t getNumberOfSimplicesInDecomposition (void) const`
- `SimplexType getSimplexFromDecomposition (const UnsignedIndex_t a_tet_number_to_get) const`
- `Volume calculateVolume (void) const`
Calculate and return volume of the tet.
- `Volume calculateAbsoluteVolume (void) const`
Calculate and return signed volume of the tet. See Owkes & Desjardins, JCP, 2014.
- `double calculateSign (void) const`
Calculate sign for the tet See Owkes & Desjardins, JCP, 2014.
- `Pt calculateCentroid (void) const`
Calculate and return centroid of the tet.
- `VolumeMoments calculateMoments () const`
Calculate and return volume weighted VolumeMoments.

Private Member Functions

- const Derived & **getDerived** (void) const

The documentation for this class was generated from the following file:

- polyhedron_moments_calculation.h

6.768 f_dividedpolygon_class::printtoscreen Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_printtoscreen** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.769 f_planarlocalizer_class::printtoscreen Interface Reference

Public Member Functions

- subroutine **planarlocalizer_class_printtoscreen** (this)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.770 f_planarseparator_class::printtoscreen Interface Reference

Public Member Functions

- subroutine **planarseparator_class_printtoscreen** (this)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.771 f_polygon_class::printtoscreen Interface Reference

Public Member Functions

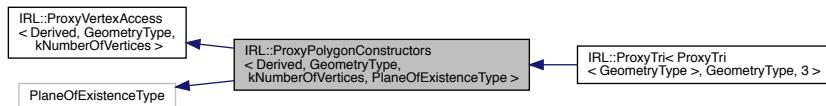
- subroutine **polygon_class_printtoscreen** (this)

The documentation for this interface was generated from the following file:

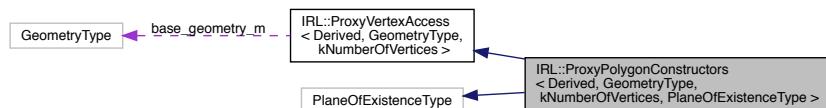
- [f_polygon_class.f90](#)

6.772 IRL::ProxyPolygonConstructors< Derived, GeometryType, kNumberOfVertices, PlaneOfExistenceType > Class Template Reference

Inheritance diagram for IRL::ProxyPolygonConstructors< Derived, GeometryType, kNumberOfVertices, PlaneOfExistenceType >:



Collaboration diagram for IRL::ProxyPolygonConstructors< Derived, GeometryType, kNumberOfVertices, PlaneOfExistenceType >:



Public Member Functions

- **ProxyPolygonConstructors** (const GeometryType &a_base_geometry, std::initializer_list< UnsignedIndex_t > a_list)
- **ProxyPolygonConstructors** (const GeometryType &a_base_geometry, std::initializer_list< UnsignedIndex_t > a_list, const [Plane](#) &a_plane_of_existence)
- **ProxyPolygonConstructors** (const GeometryType &a_base_geometry, std::array< UnsignedIndex_t, kNumberOfVertices > a_list)
- **ProxyPolygonConstructors** (const GeometryType &a_base_geometry, std::array< UnsignedIndex_t, kNumberOfVertices > a_list, const [Plane](#) &a_plane_of_existence)

Static Public Member Functions

- static Derived **fromNoExistencePlane** (const GeometryType &a_base_geometry, std::initializer_list< UnsignedIndex_t > a_list)
- static Derived **fromNoExistencePlane** (const GeometryType &a_base_geometry, std::array< UnsignedIndex_t, kNumberOfVertices > a_list)

Private Member Functions

- **ProxyPolygonConstructors** (const GeometryType &a_base_geometry, std::initializer_list< UnsignedIndex_t > a_list, Plane *a_plane_of_existence)
- **ProxyPolygonConstructors** (const GeometryType &a_base_geometry, std::array< UnsignedIndex_t, kNumberOfVertices > a_list, Plane *a_plane_of_existence)

Additional Inherited Members

The documentation for this class was generated from the following file:

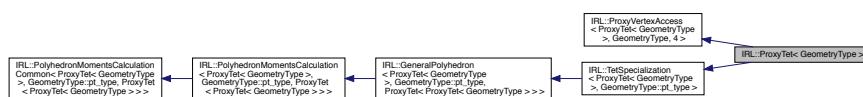
- polygon_storage_types.h

6.773 IRL::ProxyTet< GeometryType > Class Template Reference

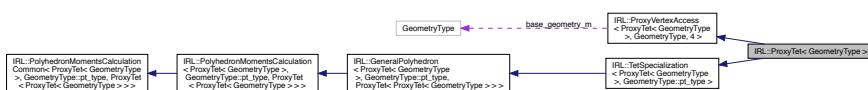
A tet from a collection of 4 points.

```
#include <tet.h>
```

Inheritance diagram for IRL::ProxyTet< GeometryType >:



Collaboration diagram for IRL::ProxyTet< GeometryType >:



Additional Inherited Members

6.773.1 Detailed Description

```
template<class GeometryType>
class IRL::ProxyTet< GeometryType >
```

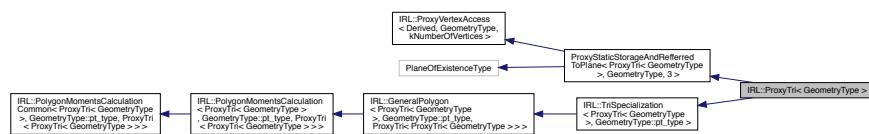
A tet from a collection of 4 points.

The documentation for this class was generated from the following file:

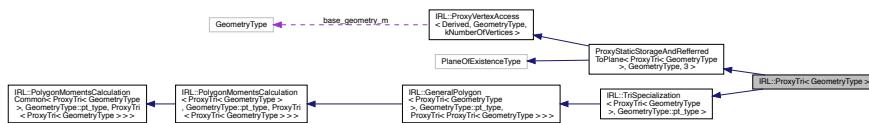
- general_polyhedron.h

6.774 IRL::ProxyTri< GeometryType > Class Template Reference

Inheritance diagram for IRL::ProxyTri< GeometryType >:



Collaboration diagram for IRL::ProxyTri< GeometryType >:



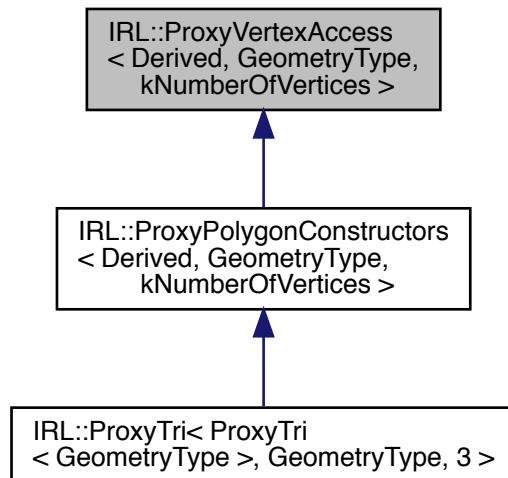
Additional Inherited Members

The documentation for this class was generated from the following file:

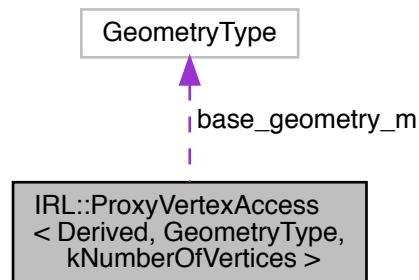
- tri.h

6.775 IRL::ProxyVertexAccess< Derived, GeometryType, kNumberOfVertices > Class Template Reference

Inheritance diagram for IRL::ProxyVertexAccess< Derived, GeometryType, kNumberOfVertices >:



Collaboration diagram for IRL::ProxyVertexAccess< Derived, GeometryType, kNumberOfVertices >:



Public Types

- using `VertexType` = typename `GeometryType::pt_type`

Public Member Functions

- **ProxyVertexAccess** (const GeometryType &a_base_geometry, std::initializer_list< UnsignedIndex_t > a_list)
- **ProxyVertexAccess** (const GeometryType &a_base_geometry, std::array< UnsignedIndex_t, kNumberOfVertices > a_list)
- const VertexType & **access** (const UnsignedIndex_t a_index) const
- const std::array< UnsignedIndex_t, kNumberOfVertices > & **getIndexMapping** (void) const
- LargeOffsetIndex_t **getSerializedSize** (void) const

Return size of the serialized Polyhedron.
- void **serialize** (ByteBuffer *a_buffer) const

Serialize and pack the Polyhedron, copying the points. This will need to be unpacked into an object of this type that has its own storage.
- **ProxyVertexAccess** (const ProxyVertexAccess &a_other)
- **ProxyVertexAccess** & **operator=** (const ProxyVertexAccess &a_other)
- **ProxyVertexAccess** (ProxyVertexAccess &&a_other)
- **ProxyVertexAccess** & **operator=** (ProxyVertexAccess &&a_other)

Static Public Member Functions

- static constexpr UnsignedIndex_t **getNumberOfVerticesInObject** (void)

Return the number of vertices in this polygon.

Static Public Attributes

- static constexpr UnsignedIndex_t **number_of_vertices** = kNumberOfVertices

Private Attributes

- const GeometryType & **base_geometry_m**
- std::array< UnsignedIndex_t, kNumberOfVertices > **index_mapping_m**

The documentation for this class was generated from the following file:

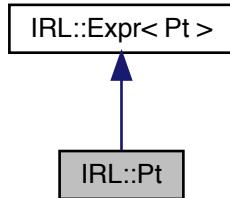
- proxy_vertex_access.h

6.776 IRL::Pt Class Reference

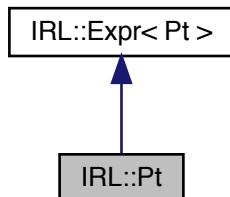
A point in 3D space.

```
#include <pt.h>
```

Inheritance diagram for IRL::Pt:



Collaboration diagram for IRL::Pt:



Public Types

- using **value_type** = double

Public Member Functions

- **Pt** (void)=default
The default constructor, performs NO INITIALIZATION.
- **constexpr Pt** (const double a_x, const double a_y, const double a_z)
A constructor that initializes the point given the x,y, and z locations.
- **operator Vec3** (void)
- **Pt & operator=** (const double a_value)
Assignment of point to a constant double.
- **Pt & operator*=(** const double a_value)
Multiply point by constant value.
- **Pt & getPt** (void)
Functions necessary to have consistent interface to PtWithData.
- **const Pt & getPt** (void) const

- template<class E >
Pt (const **Expr**< E > &a_expr)
- template<class E >
Pt (**Expr**< E > &&a_expr)
- template<class E >
Pt & **operator=** (const **Expr**< E > &a_expr)
- template<class E >
Pt & **operator=** (**Expr**< E > &&a_expr)
- double & **operator[]** (const UnsignedIndex_t a_d)
Return reference to a_d index of the point.
- const double & **operator[]** (const UnsignedIndex_t a_d) const
Const version of operator[].
- double & **x** (void)
Return value of loc_m[0] (x location).
- double & **y** (void)
Return value of loc_m[1] (y location).
- double & **z** (void)
Return value of loc_m[2] (z location).
- const double & **x** (void) const
Const version of x().
- const double & **y** (void) const
Const version of y().
- const double & **z** (void) const
Const version of z().
- UnsignedIndex_t **maxComponent** (void) const
Return the index of the dimension with maximum magnitude.
- **Pt** **operator-** (void)
- **Pt** & **operator+=** (const **Pt** &a_rhs)
Increment this point by another point.
- **Pt** & **operator-=** (const **Pt** &a_rhs)
Decrement this point by another point.
- **Pt** & **operator/** (const double a_rhs)
Divide each point location by the constant a_rhs.
- **Pt** & **operator/=** (const double a_rhs)
Divide each point location by the constant a_rhs.
- LargeOffsetIndex_t **getSerializedSize** (void) const
Return size of the serialized point class in bytes.
- void **serialize** (ByteBuffer *a_buffer) const
Serialize the point and store in the ByteBuffer.
- void **unpackSerialized** (ByteBuffer *a_buffer)
Unpack the serialized point and store.
- **~Pt** (void)=default
Default destructor.

Static Public Member Functions

- static **Pt** **fromEdgeIntersection** (const **Pt** &a_pt_0, const double a_dist_0, const **Pt** &a_pt_1, const double a_dist_1)
Performs the interpolation on an edge between points with known distances to the plane to calculate the intersection point.
- static **Pt** **fromRawDoublePointer** (const double *a_loc)
- static **Pt** **fromArray** (const std::array< double, 3 > &a_loc)
- static constexpr **Pt** **fromScalarConstant** (const double a_value)
- static **Pt** **fromVec3** (const **Vec3** &a_vec)

Private Member Functions

- `constexpr Pt (const double *a_loc)`
A constructor that initializes the point given 3 elements in the array a_loc.
- `constexpr Pt (const std::array< double, 3 > &a_loc)`
A constructor that initializes the point given 3 elements in the array a_loc.
- `constexpr Pt (const double a_value)`
Construct from a scalar constant value.
- `Pt (const Vec3 &a_vec)`
Construct from a Vec3 MathVector.

Private Attributes

- `std::array< double, 3 > loc_m`
x,y,z (0,1,2) location of the point.

Friends

- `constexpr UnsignedIndex_t size (const Pt &x)`
Length of point vector.

6.776.1 Detailed Description

A point in 3D space.

6.776.2 Member Function Documentation

6.776.2.1 fromEdgeIntersection()

```
static Pt IRL::Pt::fromEdgeIntersection (
    const Pt & a_pt_0,
    const double a_dist_0,
    const Pt & a_pt_1,
    const double a_dist_1 ) [static]
```

Performs the interpolation on an edge between points with known distances to the plane to calculate the intersection point.

This function takes two vertices of an edge and the corresponding signed distances to an intersecting plane. The point of intersection (where dist=0) is then calculated and returned.

Parameters

in	<code>a_point_1</code>	First point on the edge.
in	<code>a_dist_1</code>	Corresponding signed distance from intersecting plane for a_point_1.
in	<code>a_point_2</code>	Second point on the edge.
in	<code>a_dist_2</code>	Corresponding signed distance from intersecting plane for a_point_2.

The documentation for this class was generated from the following file:

- pt.h

6.777 IRL::PtList< VertexType, kMaxNumberOfPts > Class Template Reference

Public Types

- using **iterator** = typename std::array< VertexType, kMaxNumberOfPts >::iterator
- using **const_iterator** = typename std::array< VertexType, kMaxNumberOfPts >::const_iterator

Public Member Functions

- **PtList** (void)=default
Default constructor.
- **PtList** (std::initializer_list< VertexType > a_list)
- const VertexType * **getPtList** (void) const
Return const pointer to the vertices.
- VertexType & **operator[]** (const UnsignedIndex_t a_index)
Access through overloaded operator[].
- const VertexType & **operator[]** (const UnsignedIndex_t a_index) const
Const access through overloaded operator[].
- **IRL::Pt getLowerLimits** (void) const
Return a point for the lower limits of the polygon in 3D space.
- **IRL::Pt getUpperLimits** (void) const
Return a point for the upper limits of the polygon in 3D space.
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept
- LargeOffsetIndex_t **getSerializedSize** (void) const
Return size of the serialized Ptlist.
- void **serialize** (ByteBuffer *a_buffer) const
Serialize and pack the points.
- void **unpackSerialized** (ByteBuffer *a_buffer)
Unpack the points and store.
- **~PtList** (void)=default
Default destructor.

Static Public Member Functions

- static **PtList fromRawPtPointer** (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts)
- static **PtList fromRawDoublePointer** (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs)
- static constexpr UnsignedIndex_t **getNumberOfPts** (void)
Return the number of vertices in this polygon.

Protected Member Functions

- `constexpr PtList (const VertexType &a_pt0, const VertexType &a_pt1, const VertexType &a_pt2, const VertexType &a_pt3)`
`Construct a tetrahedron.`
- `PtList (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts)`
`Construct n-pts from array of points.`
- `PtList (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs)`
`Construct n-pts from array of doubles.`

Private Attributes

- `std::array< VertexType, kMaxNumberOfPts > pt_list_m`

Friends

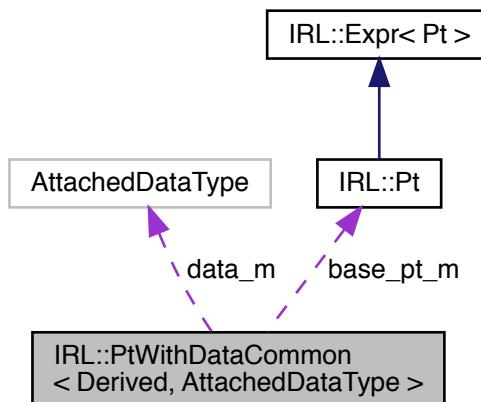
- class `Polyhedron< VertexType, kMaxNumberOfPts >`

The documentation for this class was generated from the following file:

- `pt_list.h`

6.778 IRL::PtWithDataCommon< Derived, AttachedDataType > Class Template Reference

Collaboration diagram for `IRL::PtWithDataCommon< Derived, AttachedDataType >`:



Public Types

- using **contained_type** = AttachedDataType

Public Member Functions

- PtWithDataCommon** (const **Pt** &a_pt, const AttachedDataType &a_data)
- PtWithDataCommon** (const **Pt** &a_pt)
- double & **operator[]** (const UnsignedIndex_t a_index)
Provides access to underlying base_pt location.
- const double & **operator[]** (const UnsignedIndex_t a_index) const
- Pt** & **getPt** (void)
- const **Pt** & **getPt** (void) const
- AttachedDataType & **getData** (void)
- const AttachedDataType & **getData** (void) const

Private Member Functions

- Derived & **getDerived** (void)
- const Derived & **getDerived** (void) const

Private Attributes

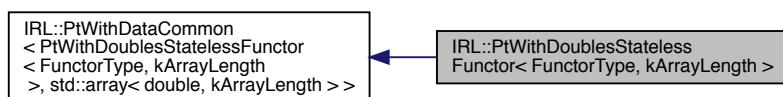
- Pt** **base_pt_m**
- AttachedDataType **data_m**

The documentation for this class was generated from the following file:

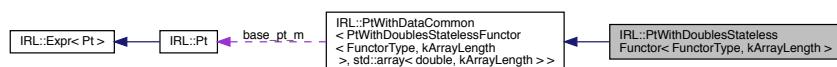
- pt_with_data.h

6.779 IRL::PtWithDoublesStatelessFunctor< FunctorType, kArrayLength > Class Template Reference

Inheritance diagram for IRL::PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >:



Collaboration diagram for IRL::PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >:



Public Member Functions

- `PtWithDoublesStatelessFunctor` (const `Pt` &a_pt)
- `PtWithDoublesStatelessFunctor` & `operator=` (const `Pt` &a_pt)
- `PtWithDoublesStatelessFunctor` & `operator+=` (const `PtWithDoublesStatelessFunctor` &a_other_pt)
- `PtWithDoublesStatelessFunctor` & `operator/=` (const double a_double)

Static Public Member Functions

- static `PtWithDoublesStatelessFunctor` `fromEdgeIntersection` (const `PtWithDoublesStatelessFunctor` &a_pt_0, const double a_dist_0, const `PtWithDoublesStatelessFunctor` &a_pt_1, const double a_dist_1)

Static Public Attributes

- static constexpr UnsignedIndex_t `data_length` = kArrayLength

Private Types

- using `ArrayType` = std::array< double, kArrayLength >

Additional Inherited Members

The documentation for this class was generated from the following file:

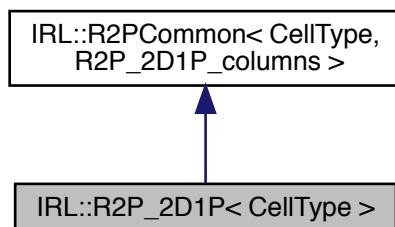
- `pt_with_data.h`

6.780 IRL::R2P_2D1P< CellType > Class Template Reference

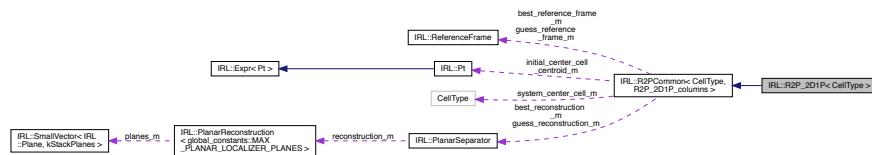
R2P class for reconstructions in 2 dimensions with 1 plane (hence 2D1P)

```
#include <r2p_optimization.h>
```

Inheritance diagram for IRL::R2P_2D1P< CellType >:



Collaboration diagram for IRL::R2P_2D1P< CellType >:



Public Types

- using **cell_type** = CellType

Public Member Functions

- R2P_2D1P** (void)=default
Default constructor.
- ~R2P_2D1P** (void)=default
Default destructor.
- PlanarSeparator solve** (const **R2PNeighborhood< CellType >** &a_neighborhood, const **PlanarSeparator** &a_reconstruction)
Perform optimization and return planar separator.
- void setup** (const **R2PNeighborhood< CellType >** &a_neighborhood, const **PlanarSeparator** &a_reconstruction)
Initialize simulation parameters that will be needed during the optimization.
- void allocateMatrices** (const UnsignedIndex_t a_neighborhood_size)
- void updateGuess** (const Eigen::Matrix< double, **columns_m**, 1 > *const a_delta)
Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.
- void updateBestGuess** (void)
Save current guess as best and update all internally saved variables such as reference frame, beta, etc.
- bool minimumReached** (const Eigen::Matrix< double, **columns_m**, 1 > &delta)
Returns whether the minimum is reached.
- auto getDefaultInitialDelta** (void) -> const Eigen::Matrix< double, **columns_m**, 1 > &
Return initial delta to be used in optimization.

Static Public Attributes

- static constexpr UnsignedIndex_t columns_m = R2P_2D1P_columns**
Number of columns for R2P_2D1P optimization.

Private Member Functions

- UnitQuaternion getDeltaRotationQuat** (const **ReferenceFrame** &a_reference_frame, const Eigen::Matrix< double, **columns_m**, 1 > &a_delta)
Return rotation for R2P dictated by elements in a_delta.
- PlanarSeparator getReconstructionFromR2PParam** (const **ReferenceFrame** &a_reference_frame)
Return a reconstruction from R2P parameters.

Private Attributes

- Eigen::Matrix< double, `columns_m`, 1 > `initial_delta_m`
Initial delta to supply to Levenberg Marquardt to calculate Jacobian.

Additional Inherited Members

6.780.1 Detailed Description

```
template<class CellType>
class IRL::R2P_2D1P< CellType >
```

R2P class for reconstructions in 2 dimensions with 1 plane (hence 2D1P)

6.780.2 Member Function Documentation

6.780.2.1 getDeltaRotationQuat()

```
template<class CellType >
UnitQuaternion IRL::R2P_2D1P< CellType >::getDeltaRotationQuat (
    const ReferenceFrame & a_reference_frame,
    const Eigen::Matrix< double, columns_m, 1 > & a_delta ) [private]
```

Return rotation for R2P dictated by elements in a_delta.

The rotation order is:

- Rotate by a_delta(0) radians around a_reference_frame[0]

6.780.2.2 updateGuess()

```
template<class CellType >
void IRL::R2P_2D1P< CellType >::updateGuess (
    const Eigen::Matrix< double, columns_m, 1 > *const a_delta )
```

Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.

Delta may need to be modified as well when placing the the planes requires reprojection of the distances onto a volume-conserving distance.

The documentation for this class was generated from the following file:

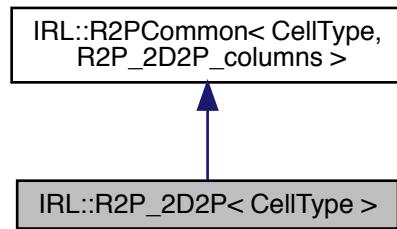
- [r2p_optimization.h](#)

6.781 IRL::R2P_2D2P< CellType > Class Template Reference

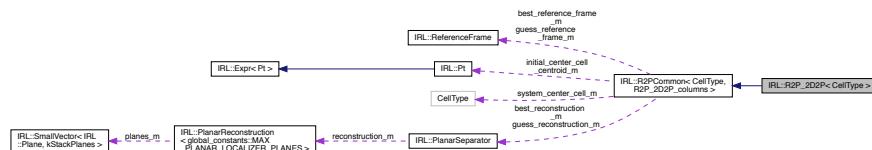
R2P class for reconstructions in 2 dimensions with 2 plane (hence 2D2P)

```
#include <r2p_optimization.h>
```

Inheritance diagram for IRL::R2P_2D2P< CellType >:



Collaboration diagram for IRL::R2P_2D2P< CellType >:



Public Types

- using **cell_type** = CellType

Public Member Functions

- **R2P_2D2P** (void)=default
Default constructor.
- **~R2P_2D2P** (void)=default
Default destructor.
- **PlanarSeparator solve** (const **R2PNeighborhood**< CellType > &a_neighborhood, const **PlanarSeparator** &a_reconstruction)
Perform optimization and return planar separator.
- **void setup** (const **R2PNeighborhood**< CellType > &a_neighborhood, const **PlanarSeparator** &a_reconstruction)
Initialize simulation parameters that will be needed during the optimization.
- **void allocateMatrices** (const UnsignedIndex_t a_neighborhood_size)
- **void updateGuess** (Eigen::Matrix< double, **columns_m**, 1 > *a_delta)

- Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.*
- void `updateBestGuess` (void)

Save current guess as best and update all internally saved variables such as reference frame, beta, etc.
 - bool `minimumReached` (const Eigen::Matrix< double, `columns_m`, 1 > &`delta`)

Returns whether the minimum is reached.
 - auto `getDefaultInitialDelta` (void) -> const Eigen::Matrix< double, `columns_m`, 1 > &

Return initial delta to be used in optimization.
 - const double & `getBestBeta` (void)

Return best beta_m value.

Static Public Attributes

- static constexpr UnsignedIndex_t `columns_m` = `R2P_2D2P_columns`

Number of columns for R2P_2D2P optimization.

Private Member Functions

- `UnitQuaternion getDeltaRotationQuat` (const `ReferenceFrame` &`a_reference_frame`, const Eigen::Matrix< double, `columns_m`, 1 > &`a_delta`)

Return rotation for R2P dictated by elements in a_delta.
- `PlanarSeparator getReconstructionFromR2PParam` (const `ReferenceFrame` &`a_reference_frame`, const double `a_beta`, const double *`a_distances`)

Return a reconstruction from R2P parameters.

Private Attributes

- Eigen::Matrix< double, `columns_m`, 1 > `initial_delta_m`

Initial delta to supply to Levenberg Marquardt to calculate Jacobian.
- double `guess_beta_m`

Current guess for beta, the rotation from shared normal to two plane normals.
- double `guess_distances_m` [2]

Current guess for distance to each plane.
- double `best_beta_m`

Current best value for beta, the rotation from shared normal to two plane normals, that gives the least error.
- double `best_distances_m` [2]

Current best values for distance to each plane that gives the least error.

Additional Inherited Members

6.781.1 Detailed Description

```
template<class CellType>
class IRL::R2P_2D2P< CellType >
```

R2P class for reconstructions in 2 dimensions with 2 plane (hence 2D2P)

6.781.2 Member Function Documentation

6.781.2.1 getDeltaRotationQuat()

```
template<class CellType >
UnitQuaternion IRL::R2P_2D2P< CellType >::getDeltaRotationQuat (
    const ReferenceFrame & a_reference_frame,
    const Eigen::Matrix< double, columns_m, 1 > & a_delta ) [private]
```

Return rotation for R2P dictated by elements in a_delta.

The rotation order is:

- Rotate by a_delta(0) radians around a_reference_frame[0]

6.781.2.2 updateGuess()

```
template<class CellType >
void IRL::R2P_2D2P< CellType >::updateGuess (
    Eigen::Matrix< double, columns_m, 1 > * a_delta )
```

Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.

Delta may need to be modified as well when placing the the planes requires reprojection of the distances onto a volume-conserving distance.

The documentation for this class was generated from the following file:

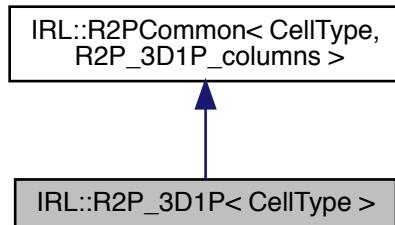
- [r2p_optimization.h](#)

6.782 IRL::R2P_3D1P< CellType > Class Template Reference

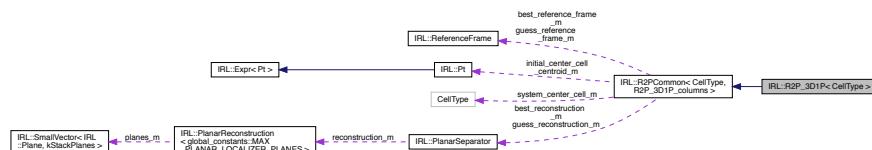
R2P class for reconstructions in 3 dimensions with 1 plane (hence 3D1P)

```
#include <r2p_optimization.h>
```

Inheritance diagram for IRL::R2P_3D1P< CellType >:



Collaboration diagram for IRL::R2P_3D1P< CellType >:



Public Types

- using **cell_type** = CellType

Public Member Functions

- R2P_3D1P** (void)=default
Default constructor.
- ~R2P_3D1P** (void)=default
Default destructor.
- PlanarSeparator solve** (const **R2P Neighborhood**< CellType > &a_neighborhood, const **PlanarSeparator** &a_reconstruction)
Perform optimization and return planar separator.
- void setup** (const **R2P Neighborhood**< CellType > &a_neighborhood, const **PlanarSeparator** &a_reconstruction)
Initialize simulation parameters that will be needed during the optimization.
- void allocateMatrices** (const UnsignedIndex_t a_neighborhood_size)
- void updateGuess** (const Eigen::Matrix< double, **columns_m**, 1 > *const a_delta)
Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.
- void updateBestGuess** (void)
Save current guess as best and update all internally saved variables such as reference frame, beta, etc.
- bool minimumReached** (const Eigen::Matrix< double, **columns_m**, 1 > &delta)
Returns whether the minimum is reached.
- auto getDefaultInitialDelta** (void) -> const Eigen::Matrix< double, **columns_m**, 1 > &
Return initial delta to be used in optimization.

Static Public Attributes

- static constexpr UnsignedIndex_t `columns_m` = `R2P_3D1P_columns`
Number of columns for R2P_3D1P optimization.

Private Member Functions

- `UnitQuaternion getDeltaRotationQuat` (const `ReferenceFrame` &`a_reference_frame`, const `Eigen::Matrix<double, columns_m, 1>` &`a_delta`)
Return rotation for R2P dictated by elements in a_delta.
- `PlanarSeparator getReconstructionFromR2PParam` (const `ReferenceFrame` &`a_reference_frame`)
Return a reconstruction from R2P parameters.

Private Attributes

- `Eigen::Matrix<double, columns_m, 1>` `initial_delta_m`
Initial delta to supply to Levenberg Marquardt to calculate Jacobian.

Additional Inherited Members

6.782.1 Detailed Description

```
template<class CellType>
class IRL::R2P_3D1P< CellType >
```

R2P class for reconstructions in 3 dimensions with 1 plane (hence 3D1P)

6.782.2 Member Function Documentation

6.782.2.1 getDeltaRotationQuat()

```
template<class CellType >
UnitQuaternion IRL::R2P_3D1P< CellType >::getDeltaRotationQuat (
    const ReferenceFrame & a_reference_frame,
    const Eigen::Matrix< double, columns_m, 1 > & a_delta ) [private]
```

Return rotation for R2P dictated by elements in a_delta.

The rotation order is:

- Rotate by `a_delta(0)` radians around `a_reference_frame[0]`
- Rotate by `a_delta(1)` radians around `a_reference_frame[1]`

6.782.2.2 updateGuess()

```
template<class CellType >
void IRL::R2P_3D1P< CellType >::updateGuess (
    const Eigen::Matrix< double, columns_m, 1 > *const a_delta )
```

Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.

Delta may need to be modified as well when placing the the planes requires reprojection of the distances onto a volume-conserving distance.

The documentation for this class was generated from the following file:

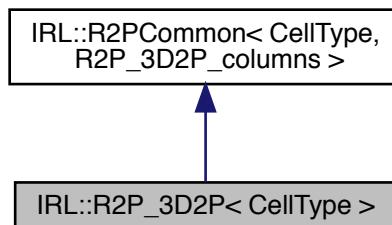
- [r2p_optimization.h](#)

6.783 IRL::R2P_3D2P< CellType > Class Template Reference

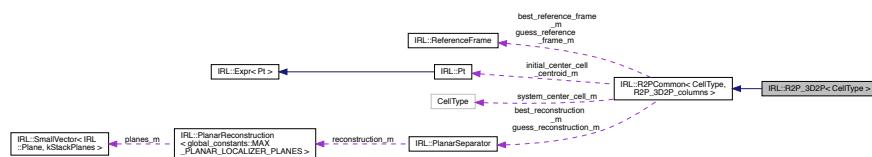
R2P class for reconstructions in 3 dimensions with 2 plane (hence 3D2P)

```
#include <r2p_optimization.h>
```

Inheritance diagram for IRL::R2P_3D2P< CellType >:



Collaboration diagram for IRL::R2P_3D2P< CellType >:



Public Types

- using **cell_type** = CellType

Public Member Functions

- `R2P_3D2P (void)=default`
Default constructor.
- `~R2P_3D2P (void)=default`
Default destructor.
- `PlanarSeparator solve (const R2PNeighborhood< CellType > &a_neighborhood, const PlanarSeparator &a_reconstruction)`
Perform optimization and return planar separator.
- `void setup (const R2PNeighborhood< CellType > &a_neighborhood, const PlanarSeparator &a_reconstruction)`
Initialize simulation parameters that will be needed during the optimization.
- `void allocateMatrices (const UnsignedIndex_t a_neighborhood_size)`
- `void updateGuess (Eigen::Matrix< double, columns_m, 1 > *a_delta)`
Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.
- `void updateBestGuess (void)`
Save current guess as best and update all internally saved variables such as reference frame, beta, etc.
- `bool minimumReached (const Eigen::Matrix< double, columns_m, 1 > &delta)`
Returns whether the minimum is reached.
- `auto getDefaultInitialDelta (void) -> const Eigen::Matrix< double, columns_m, 1 > &`
Return initial delta to be used in optimization.
- `const double & getBestBeta (void)`
Return best beta_m value.

Static Public Attributes

- `static constexpr UnsignedIndex_t columns_m = R2P_3D2P_columns`
Number of columns for R2P_3D2P optimization.

Private Member Functions

- `UnitQuaternion getDeltaRotationQuat (const ReferenceFrame &a_reference_frame, const Eigen::Matrix< double, columns_m, 1 > &a_delta)`
Return rotation for R2P dictated by elements in a_delta.
- `PlanarSeparator getReconstructionFromR2PParam (const ReferenceFrame &a_reference_frame, const double a_beta, const double *a_distances)`
Return a reconstruction from R2P parameters.

Private Attributes

- `Eigen::Matrix< double, columns_m, 1 > initial_delta_m`
Initial delta to supply to Levenberg Marquardt to calculate Jacobian.
- `double guess_beta_m`
Current guess for beta, the rotation from shared normal to two plane normals.
- `double guess_distances_m [2]`
Current guess for distance to each plane.
- `double best_beta_m`
Current best value for beta, the rotation from shared normal to two plane normals, that gives the least error.
- `double best_distances_m [2]`
Current best values for distance to each plane that gives the least error.

Additional Inherited Members

6.783.1 Detailed Description

```
template<class CellType>
class IRL::R2P_3D2P< CellType >
```

R2P class for reconstructions in 3 dimensions with 2 plane (hence 3D2P)

6.783.2 Member Function Documentation

6.783.2.1 getDeltaRotationQuat()

```
template<class CellType >
UnitQuaternion IRL::R2P_3D2P< CellType >::getDeltaRotationQuat (
    const ReferenceFrame & a_reference_frame,
    const Eigen::Matrix< double, columns_m, 1 > & a_delta ) [private]
```

Return rotation for R2P dictated by elements in a_delta.

The rotation order is:

- Rotate by a_delta(0) radians around a_reference_frame[0]
- Rotate by a_delta(1) radians around a_reference_frame[1]
- Rotate by a_delta(2) radians around a_reference_frame[2]

6.783.2.2 updateGuess()

```
template<class CellType >
void IRL::R2P_3D2P< CellType >::updateGuess (
    Eigen::Matrix< double, columns_m, 1 > * a_delta )
```

Updates current guess reconstruction using a_delta and calculates the new reconstruction as well as stores weighted guess_value vector in guess_values_m.

Delta may need to be modified as well when placing the the planes requires reprojection of the distances onto a volume-conserving distance.

The documentation for this class was generated from the following file:

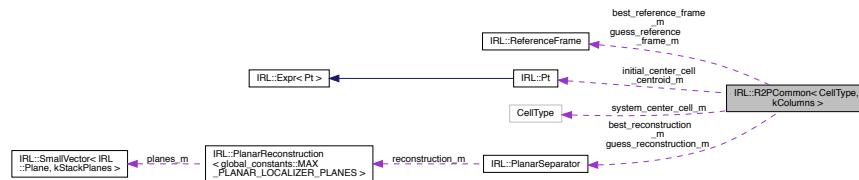
- [r2p_optimization.h](#)

6.784 IRL::R2PCommon< CellType, kColumns > Class Template Reference

Class to contain data and methods that will be used in all of the specific R2Optimization.

```
#include <r2p_optimization.h>
```

Collaboration diagram for IRL::R2PCommon< CellType, kColumns >:



Public Member Functions

- template<class R2PType >
PlanarSeparator runOptimization (R2PType *a_ptr_to_R2P_object, const R2PNeighborhood< CellType > &a_neighborhood_geometry, const PlanarSeparator &a_reconstruction)
Class that takes a pointer to a R2P object and executes the optimization, returning the found optimal reconstruction.
- **PlanarSeparator getFinalReconstruction** (void)
Return the final reconstruction to be used.
- Eigen::Matrix< double, Eigen::Dynamic, 1 > **calculateVectorError** (void)
Calculate the vector error a_correct_vector - a_attempt_vector where both vectors are weighted geometry vectors.
- bool **errorTooHigh** (const double a_error)
Return a boolean stating whether the value is too high or not.
- bool **iterationTooHigh** (const UnsignedIndex_t a_iteration)
Return a boolean stating whether maximum iterations has been exceeded.
- void **increaseLambda** (double *a_lambda)
Increase lambda to decrease step size.
- void **decreaseLambda** (double *a_lambda)
Decrease lambda to increase step size.
- bool **shouldComputeJacobian** (const UnsignedIndex_t a_iteration, const UnsignedIndex_t a_last_jacobian)
Return whether Jacobian should be computed, allowing delayed re-evaluation of the Jacobian.
- double **calculateScalarError** (void)
Calculate the error ||a_correct_vector - a_attempt_vector||^2 where both vectors are weighted geometry vectors.
- void **saveBestGuess** (void)
Save current guess_values_m and current_reconstruction_m.
- Eigen::Matrix< double, Eigen::Dynamic, 1 > **calculateChangeInGuess** (void)
Calculate and return vector of differences between guess_values_m and best_values_m. Used in Jacobian calculation.
- const PlanarSeparator & **getBestReconstruction** (void)
Return the best reconstruction found during the optimization procedure.
- const PlanarSeparator & **getGuessReconstruction** (void)
Return a const reference to the guess reconstruction.
- const CellType & **getCell** (const UnsignedIndex_t a_cell)
Return const reference to cells_to_cut.
- const ReferenceFrame & **getBestReferenceFrame** (void)

- `const ReferenceFrame & getGuessReferenceFrame (void)`
Return best guess reference frame.
- `void fillGeometryAndWeightVectors (const R2PNeighborhood< CellType > &a_neighborhood, const double distance_multiplier, const double volume_weight_switch)`
Initial filling of correct geometry and weighting vectors during setup.
- `void setRelativelImportanceBetweenWeights (double a_importance_of_liquid_volume_fraction, double a_importance_of_liquid_centroid_relative_to_gas, double a_importance_of_centroid, double a_importance_of_surface_area)`
Adjusts weight vectors so that the magnitude sums to one and satisfies the given magnitudes of the specific weights.
- `void setWeightedGeometryVectorFromReconstruction (const PlanarSeparator &a_reconstruction)`
Use reconstruction to calculate weighted guess_values_m vector
- `void setWeightedGeometryVectorFromSurfaceArea (const PlanarSeparator &a_reconstruction)`
Use reconstruction to calculate weighted surface area contribution to guess_values_m.

Static Public Member Functions

- `static void addCellMomentsToGeometryVector (const SeparatedMoments< VolumeMoments > &a_moments_from_cut_cell, const Eigen::Matrix< double, Eigen::Dynamic, 1 > &a_weights, const UnsignedIndex_t a_start_index, Eigen::Matrix< double, Eigen::Dynamic, 1 > *a_geometry_vector)`
A convenience class that takes SeparatedMoments<VolumeMoments> after cutting a cell and fills geometry vector with it, starting at a_start_index.

Public Attributes

- `std::vector< CellType > cells_to_cut_m`
Cells involved in the optimization given by vertices.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > weights_m`
Weights to be applied to the corect and guess SeparatedMoments<VolumeMoments>/Surface Area.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > correct_values_m`
Weighted vector of correct values we are trying to match.
- `double volume_fraction_m`
Volume fraction in cell being reconstructed.
- `double stencil_average_volume_fraction_m`
Average volume fraction in stencil, used to set some weightings.
- `double characteristic_length_m`
Characteristic length to be used for normalization.
- `CellType system_center_cell_m`
Center cell.
- `Pt initial_center_cell_centroid_m`
Center cell centroid to be used for shifting.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > guess_values_m`
Weighted vector of guess values that we are trying to match to correct_values_m.
- `PlanarSeparator guess_reconstruction_m`
Guess reconstruction that is used to obtain guess_values_m.
- `ReferenceFrame guess_reference_frame_m`
Guess reference frame used when obtaining guess_reconstruction_m.
- `Eigen::Matrix< double, Eigen::Dynamic, 1 > best_values_m`
Weighted vector that resulted in lowest error.
- `PlanarSeparator best_reconstruction_m`
Reconstruction that has resulted in lowest error.
- `ReferenceFrame best_reference_frame_m`
Reference frame associated with the best reconstruction.

Static Public Attributes

- static constexpr double `acceptable_error_m` = $1.0e-4 * 1.0e-4$
If this->`calculateScalarError()` is less than this, exit.
- static constexpr UnsignedIndex_t `maximum_iterations_m` = 20
Maximum number of attempted iterations before exiting.
- static constexpr double `minimum_angle_change_m` = 0.0001745329
Minimum change in angle related delta below which minimum is deemed reached.
- static constexpr double `minimum_distance_change_m` = 1.0e-4
Minimum change in distance related delta below which minimum is deemed reached.
- static constexpr double `lambda_increase_m` = 5.0
Increase factor for lambda if more damping needed.
- static constexpr double `lambda_decrease_m` = 1.0 / 10.0
Decrease factor for lambda if new best solution is found.
- static constexpr UnsignedIndex_t `delay_jacobian_amount_m` = 0
Number of iterations to allow between calculating a new Jacobian.
- static constexpr double `initial_angle_m`
Initial angle to use when first calculating Jacobian, equal to 5 degrees in radians.
- static constexpr double `initial_distance_m` = 0.001
Initial distance to use when first calculating Jacobian.

Private Attributes

- friend `R2P_2D1P< CellType >`
- friend `R2P_3D1P< CellType >`
- friend `R2P_2D2P< CellType >`
- friend `R2P_3D2P< CellType >`

Friends

- template<class R2PTypeForDebug >
class R2PDebug

6.784.1 Detailed Description

```
template<class CellType, UnsignedIndex_t kColumns>
class IRL::R2PCommon< CellType, kColumns >
```

Class to contain data and methods that will be used in all of the specific R2Poptimization.

This class holds common methods and members that are used in the specific R2P optimization classes ([R2P_2D1P](#), [R2P_3D1P](#), [R2P_2D2P](#), [R2P_3D2P](#)). This includes variables and members that determine exit criterion for the optimization through Levenberg-Marquardt.

Template parameters:

- `kNcells` : Number of cells that will be included during the optimization.
- `kRows` : Number of rows in the error/guess/correct/weight vectors.
- `kColumns` : Number of columns in Jacobian matrix. Equal to number of parameters to optimize for with Levenberg-Marquardt.

6.784.2 Member Function Documentation

6.784.2.1 addCellMomentsToGeometryVector()

```
template<class CellType, UnsignedIndex_t kColumns>
static void IRL::R2PCommon< CellType, kColumns >::addCellMomentsToGeometryVector (
    const SeparatedMoments< VolumeMoments > & a_moments_from_cut_cell,
    const Eigen::Matrix< double, Eigen::Dynamic, 1 > & a_weights,
    const UnsignedIndex_t a_start_index,
    Eigen::Matrix< double, Eigen::Dynamic, 1 > * a_geometry_vector ) [static]
```

A convenience class that takes `SeparatedMoments<VolumeMoments>` after cutting a cell and fills geometry vector with it, starting at `a_start_index`.

This function is to be used to place `SeparatedMoments<VolumeMoments>` from cut cells into a geometry vector to be used by R2P. Starting at `a_start_index`, it places 7 elements (Liquid volume, Liquid Centroid, Gas Centroid) into `a_geometry_vector`, which is a `Eigen::Matrix` vector. The template `ReturnGeometry<Vector` is needed to accept `Eigen::Matrix` vectors of different fixed sizes.

Template Requirements for `ReturnGeometryVector`:

- An accessing function `operator()` that accepts one argument (an integer as an index) and will save the incoming double upon assignment.

Parameters

in	<code>a_moments_from_cut_cell</code>	<code>SeparatedMoments<VolumeMoments></code> to be added to <code>a_geometry_vector</code>
in	<code>a_weights</code>	Vector of weights to component-wise multiply with <code>a_geometry_vector</code> .
in, out	<code>a_start_index</code>	Starting location to begin adding elements to <code>a_geometry_vector</code> .
out	<code>a_geometry_vector</code>	Geometry vector to have elements added to.

6.784.2.2 fillGeometryAndWeightVectors()

```
template<class CellType, UnsignedIndex_t kColumns>
void IRL::R2PCommon< CellType, kColumns >::fillGeometryAndWeightVectors (
    const R2PNeighborhood< CellType > & a_neighborhood,
    const double distance_multiplier,
    const double volume_weight_switch )
```

Initial filling of correct geometry and weighting vectors during setup.

Assumes that Neighborhood geometry (all centroids/cells) have already been moved to be with respect to the center of the cell undergoing reconstruction (which would be at the 0,0,0 location in the stencil) and scaled so that each cell is a shifted unit cube. Upon returning, the `correct_values_m` vector will be filled with 7 entries per cell:

- Liquid volume fraction
- Liquid centroid x component
- Liquid centroid y component
- Liquid centroid z component
- Gas centroid x component
- Gas centroid y component
- Gas centroid z component

These entries will be in the "positive and descending" normal reference frame.

6.784.2.3 runOptimization()

```
template<class CellType, UnsignedIndex_t kColumns>
template<class R2PType >
PlanarSeparator IRL::R2PCommon< CellType, kColumns >::runOptimization (
    R2PType * a_ptr_to_R2P_object,
    const R2PNeighborhood< CellType > & a_neighborhood_geometry,
    const PlanarSeparator & a_reconstruction )
```

Class that takes a pointer to a R2P object and executes the optimization, returning the found optimal reconstruction.

Requirements for R2PType:

- Must be either [R2P_2D1P](#), [R2P_3D1P](#), [R2P_2D2P](#), or [R2P_3D2P](#).

Parameters

in	<i>a_ptr_to_R2P_object</i>	Pointer to R2P object that will be run through optimization.
in	<i>a_neighborhood_geometry</i>	Neighborhood geometry that will be used in the optimization to calculate error we are trying to minimize.
in	<i>a_reconstruction</i>	Initial reconstruction to start R2P optimization routine from.

6.784.2.4 setRelativeImportanceBetweenWeights()

```
template<class CellType, UnsignedIndex_t kColumns>
void IRL::R2PCommon< CellType, kColumns >::setRelativeImportanceBetweenWeights (
    double a_importance_of_liquid_volume_fraction,
    double a_importance_of_liquid_centroid_relative_to_gas,
    double a_importance_of_centroid,
    double a_importance_of_surface_area )
```

Adjusts weight vectors so that the magnitude sums to one and satisfies the given magnitudes of the specific weights.

Parameters

in	<i>a_importance_of_liquid_volume_fraction</i>	How important liquid volume fraction is in the error.
in	<i>a_importance_of_liquid_centroid_relative_to_gas</i>	How important liquid centroid is compared to the gas centroid
in	<i>a_importance_of_centroid</i>	How important centroids are compared to liquid volume fraction and surface area.
in	<i>a_importance_of_surface_area</i>	Importance of matching surface area.

6.784.3 Member Data Documentation**6.784.3.1 initial_angle_m**

```
template<class CellType, UnsignedIndex_t kColumns>
constexpr double IRL::R2PCommon<CellType, kColumns>::initial_angle_m [static]
```

Initial value:

```
= 0.001 * 0.0174533
```

Initial angle to use when first calculating Jacobian, equal to 5 degrees in radians.

The documentation for this class was generated from the following file:

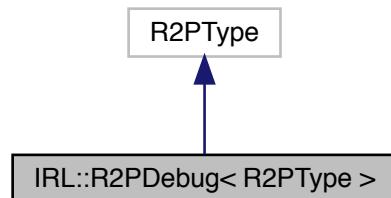
- [r2p_optimization.h](#)

6.785 IRL::R2PDebug< R2PType > Class Template Reference

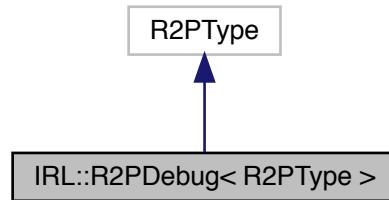
This class just calls the R2PType functions but allows debug statements to be printed.

```
#include <r2p_optimization.h>
```

Inheritance diagram for IRL::R2PDebug< R2PType >:



Collaboration diagram for IRL::R2PDebug< R2PType >:



Public Member Functions

- [R2PDebug \(void\)=default](#)
Default constructor.
- [void updateGuess \(Eigen::Matrix< double, R2PType::columns_m, 1 > *a_delta\)](#)
Calls updateGuess from the base class and stores the current reconstruction.
- [PlanarSeparator solve \(const R2PNeighborhood< typename R2PType::cell_type > &a_neighborhood, const PlanarSeparator &a_reconstruction\)](#)
Solve call for debugging R2P.
- [void updateBestGuess \(void\)](#)
Calls updateBestGuess from the base class and stores the current reconstruction.
- [PlanarSeparator getFinalReconstruction \(void\)](#)
Shadowed getFinalReconstruction call that writes out stored best reconstructions and then returns the reconstruction.
- [void writeOutPlane \(const PlanarSeparator &a_reconstruction, const std::string &a_prefix, const std::size_t a_iteration_number\)](#)
Write the ConvexPolygons in the reconstruction out to std::cout, tagged with the given iteration number.
- [void writeOutCentroidsAndWeights \(void\)](#)
Write out the centroids and weights to enable visualization of what optimization is driving towards.
- [~R2PDebug \(void\)=default](#)
Default destructor.

Private Attributes

- [std::vector< PlanarSeparator > guess_reconstruction_history](#)
Saved guess reconstructions encountered during optimization.
- [std::vector< PlanarSeparator > best_reconstruction_history](#)
Saved best reconstructions accepted during optimization.

6.785.1 Detailed Description

```
template<class R2PType>
class IRL::R2PDebug< R2PType >
```

This class just calls the R2PType functions but allows debug statements to be printed.

This class masks R2PType functions to allow calling the function and then all printing any debug information wanted. It also stores the guessed reconstructions and saved "best" reconstructions in order to reprint them out later. The printing format of this is to plot the resulting vertices of the plane polygons to be plotted by `R2P/references/r2p_history_plot.m`. To do this, the information printed to screen needs to be copied to `R2P/references/reconstruction_history.m`.

Requirements for R2PType class:

- R2PType class must meet all requirements needed by the `LevenbergMarquardt` class in `optimizers.h`. Right now, this is mainly `R2P_2D1P`, `R2P_3D1P`, `R2P_2D2P`, and `R2P_3D2P`.

The documentation for this class was generated from the following file:

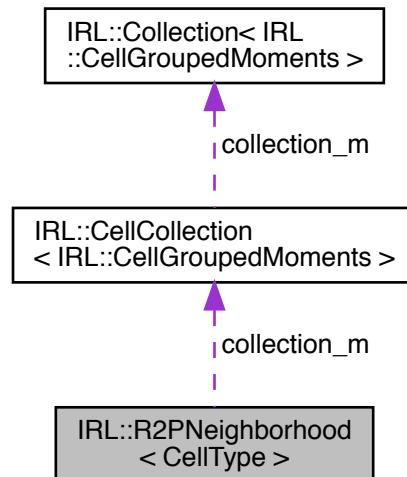
- `r2p_optimization.h`

6.786 IRL::R2PNeighborhood< CellType > Class Template Reference

Neighborhood storage used in the R2P optimization routines.

```
#include <r2p_neighborhood.h>
```

Collaboration diagram for IRL::R2PNeighborhood< CellType >:



Public Types

- using **cell_type** = CellType

Public Member Functions

- **R2P Neighborhood** (void)

Default constructor.
- void **addMember** (const CellType *a_cell, const SeparatedMoments< VolumeMoments > *a_volume_← moments)

Construct CellGroupedMoments and add to end of collection.
- void **emptyNeighborhood** (void)

Reset neighborhood size to 0.
- void **setMember** (const UnsignedIndex_t a_index, const CellType *a_cell, const SeparatedMoments< VolumeMoments > *a_volume_moments)

Construct CellGroupedMoments and place into collection.
- void **setCenterOfStencil** (const UnsignedIndex_t a_index)

Set the index for the center cell in the collection.
- void **setSurfaceArea** (const double a_surface_area)

Set the index for the center cell in the collection.
- const CellType & **getCenterCell** (void) const

Return the center cell.
- const SeparatedMoments< VolumeMoments > & **getCenterCellStoredMoments** (void) const

Return the center cell moments.
- const CGD::cell_type & **getCell** (const UnsignedIndex_t a_index) const

Return the cell stored at the index.
- const SeparatedMoments< VolumeMoments > & **getStoredMoments** (const UnsignedIndex_t a_index) const

Return moments stored at the index.
- void **resize** (const UnsignedIndex_t a_size)

Set size of the neighborhood.
- UnsignedIndex_t **size** (void) const

Get size of the collection.
- double **getSurfaceArea** (void) const

Get surface area for the center cell.
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **cend** (void) const noexcept
- **~R2P Neighborhood** (void)=default

Default destructor.

Private Types

- using **CGD** = CellGroupedMoments< CellType, SeparatedMoments< VolumeMoments > >
- using **iterator** = typename CellCollection< CGD >::iterator
- using **const_iterator** = typename CellCollection< CGD >::const_iterator

Private Member Functions

- void `checkIndex` (UnsignedIndex_t a_index) const
Make sure index is not larger than current collection size.
- void `checkCenterStencilSet` (void) const

Private Attributes

- `CellCollection< CGD > collection_m`
Collection of cells and cell moments.
- double `center_cell_surface_area_m`
brief Interface surface area for center cell in neighborhood.
- UnsignedIndex_t `center_cell_index_m`
Center stencil cell index in the list of added cells.

6.786.1 Detailed Description

```
template<class CellType>
class IRL::R2PNeighborhood< CellType >
```

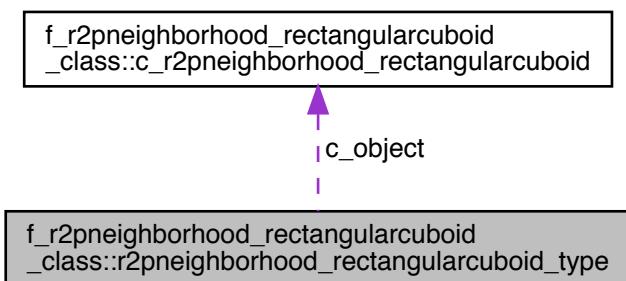
Neighborhood storage used in the R2P optimization routines.

The documentation for this class was generated from the following file:

- r2p_neighborhood.h

6.787 f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_type Type Reference

Collaboration diagram for `f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_type`:



Public Member Functions

- final `r2pneighborhood_rectangularcuboid_class_delete`

Private Attributes

- type(`c_r2pneighborhood_rectangularcuboid`), private `c_object`

The documentation for this type was generated from the following file:

- `f_r2pneighborhood_rectangularcuboid_class.f90`

6.788 `short_alloc::short_alloc< T, N, Align >::rebind< _Up >` Struct Template Reference

Public Types

- using `other` = `short_alloc< _Up, N, alignment >`

The documentation for this struct was generated from the following file:

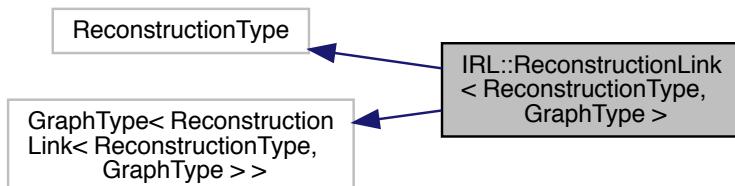
- `short_alloc.h`

6.789 `IRL::ReconstructionLink< ReconstructionType, GraphType >` Class Template Reference

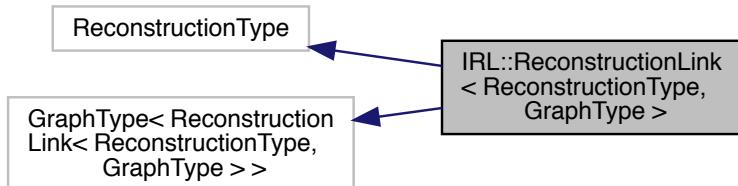
This is a class template for linking together reconstructions that are already pointers to something else. This then does not require a different constructor because we are already taking pointers of the underlying reconstructions. For classes that are currently storing their own data, it would probably be best to generate a wrapping pointer class, which can be done with `JoinedReconstruction<ClassYouWant, NullReconstruction>`.

```
#include <reconstruction_link.h>
```

Inheritance diagram for `IRL::ReconstructionLink< ReconstructionType, GraphType >`:



Collaboration diagram for IRL::ReconstructionLink< ReconstructionType, GraphType >:



Public Member Functions

- const [ReconstructionLink](#) * **getLinkingReconstructionAddress** (void) const

6.789.1 Detailed Description

```
template<class ReconstructionType, template< class NodeType > class GraphType>
class IRL::ReconstructionLink< ReconstructionType, GraphType >
```

This is a class template for linking together reconstructions that are already pointers to something else. This then does not require a different constructor because we are already taking pointers of the underlying reconstructions. For classes that are currently storing their own data, it would probably be best to generate a wrapping pointer class, which can be done with [JoinedReconstruction<ClassYouWant, NullReconstruction>](#).

The documentation for this class was generated from the following file:

- [reconstruction_link.h](#)

6.790 f_reconstructioninterface::reconstructionwithadvectedenormals Interface Reference

Public Member Functions

- subroutine **reconstructionwithadvectedenormals_listedvm_vman_rc** (a_volume_moments_list, a_neighborhood, a_two_plane_threshold, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_reconstructioninterface.f90](#)

6.791 f_reconstructioninterface::reconstructionwithadvectedenormalsdebug Interface Reference

Public Member Functions

- subroutine **reconstructionwithadvectedenormalsdebug_listedvm_vman_rc** (a_volume_moments_list, a_neighborhood, a_two_plane_threshold, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.792 f_reconstructioninterface::reconstructionwithlvira2d Interface Reference

Public Member Functions

- subroutine **reconstructionwithlvira2d_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.793 f_reconstructioninterface::reconstructionwithlvira3d Interface Reference

Public Member Functions

- subroutine **reconstructionwithlvira3d_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.794 f_reconstructioninterface::reconstructionwithmof2d Interface Reference

Public Member Functions

- subroutine **reconstructionwithmof2d_rectangularcuboid** (a_rectangular_cuboid, a_separated_volume_moments, a_planar_separator)
- subroutine **reconstructionwithmof2dgiveweights_rectangularcuboid** (a_rectangular_cuboid, a_separated_volume_moments, a_internal_weight, a_external_weight, a_planar_separator)
- subroutine **reconstructionwithmof2d_tri** (a_tri, a_separated_volume_moments, a_planar_separator)
- subroutine **reconstructionwithmof2dgiveweights_tri** (a_tri, a_separated_volume_moments, a_internal_weight, a_external_weight, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.795 f_reconstructioninterface::reconstructionwithmof3d Interface Reference

Public Member Functions

- subroutine **reconstructionwithmof3d_rectangularcuboid** (a_rectangular_cuboid, a_separated_volume_moments, a_planar_separator)
- subroutine **reconstructionwithmof3dgiveweights_rectangularcuboid** (a_rectangular_cuboid, a_separated_volume_moments, a_internal_weight, a_external_weight, a_planar_separator)
- subroutine **reconstructionwithmof3d_tet** (a_tet, a_separated_volume_moments, a_planar_separator)
- subroutine **reconstructionwithmof3dgiveweights_tet** (a_tet, a_separated_volume_moments, a_internal_weight, a_external_weight, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.796 f_reconstructioninterface::reconstructionwithr2p2d Interface Reference

Public Member Functions

- subroutine **reconstructionwithr2p2d_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.797 f_reconstructioninterface::reconstructionwithr2p2ddebug Interface Reference

Public Member Functions

- subroutine **reconstructionwithr2p2ddebug_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.798 f_reconstructioninterface::reconstructionwithr2p3d Interface Reference

Public Member Functions

- subroutine **reconstructionwithr2p3d_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.799 f_reconstructioninterface::reconstructionwithr2p3ddebug Interface Reference

Public Member Functions

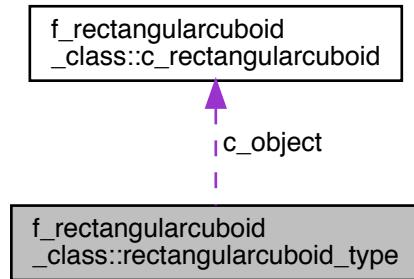
- subroutine **reconstructionwithr2p3ddebug_rc** (a_neighborhood, a_planar_separator)

The documentation for this interface was generated from the following file:

- f_reconstructioninterface.f90

6.800 f_rectangularcuboid_class::rectangularcuboid_type Type Reference

Collaboration diagram for f_rectangularcuboid_class::rectangularcuboid_type:



Public Member Functions

- final **rectangularcuboid_class_delete**

Private Attributes

- type([c_rectangularcuboid](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_rectangularcuboid_class.f90](#)

6.801 IRL::RectangularCuboidBase< VertexType > Class Template Reference

The documentation for this class was generated from the following file:

- hexahedron.h

6.802 IRL::RectangularCuboidCommon< Derived, VertexType > Class Template Reference

The documentation for this class was generated from the following file:

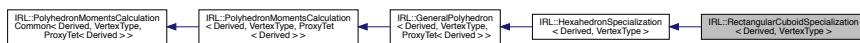
- hexahedron.h

6.803 IRL::RectangularCuboidSpecialization< Derived, VertexType > Class Template Reference

A rectangular cuboid.

```
#include <rectangular_cuboid.h>
```

Inheritance diagram for IRL::RectangularCuboidSpecialization< Derived, VertexType >:



Collaboration diagram for IRL::RectangularCuboidSpecialization< Derived, VertexType >:



Public Member Functions

- double **calculateSideLength** (const UnsignedIndex_t a_dimension) const
Calculate and return side length for a_dimension.
- **PlanarLocalizer** **getLocalizer** (void) const
Returns a planar reconstruction that is equivalent to the [RectangularCuboidCommon](#).
- double **calculateVolume** (void) const
Calculate and return the volume of the rectangular cuboid.
- Pt **calculateCentroid** (void) const
Calculate and return the centroid of the rectangular cuboid.
- **VolumeMoments** **calculateMoments** () const
Calculate and return volume weighted [VolumeMoments](#).

Additional Inherited Members

6.803.1 Detailed Description

```
template<class Derived, class VertexType>
class IRL::RectangularCuboidSpecialization< Derived, VertexType >
```

A rectangular cuboid.

The documentation for this class was generated from the following file:

- rectangular_cuboid.h

6.804 IRL::RecursiveSimplexCutting Struct Reference

The documentation for this struct was generated from the following file:

- default_cutting_method.h

6.805 IRL::ReferenceFrame Class Reference

A reference frame with three normals (1 for each direction in 3D space).

```
#include <reference_frame.h>
```

Public Member Functions

- [ReferenceFrame \(void\)=default](#)
Default constructor.
- [ReferenceFrame \(const Normal &a_axis_0, const Normal &a_axis_1, const Normal &a_axis_2\)](#)
Construct given 3 normals.
- [Normal & operator\[\] \(const UnsignedIndex_t a_axis\)](#)
Overload operator[] for access.
- [const Normal & operator\[\] \(const UnsignedIndex_t a_axis\) const](#)
Const version of overload operator[] for access.
- [~ReferenceFrame \(void\)=default](#)
Default destructor.

Private Attributes

- [std::array< Normal, 3 > axis_m](#)
Three orthonormal vectors making up the reference frame.

6.805.1 Detailed Description

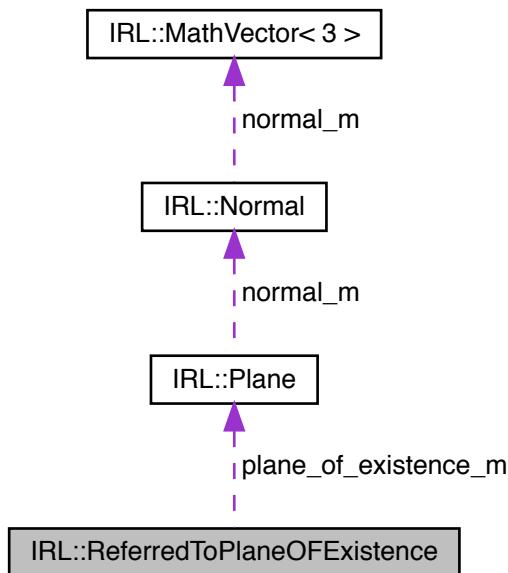
A reference frame with three normals (1 for each direction in 3D space).

The documentation for this class was generated from the following file:

- reference_frame.h

6.806 IRL::ReferredToPlaneOFExistence Class Reference

Collaboration diagram for IRL::ReferredToPlaneOFExistence:



Public Member Functions

- `ReferredToPlaneOFExistence (const Plane &a_plane_of_existence)`
- `ReferredToPlaneOFExistence (const Plane *a_plane_of_existence)`
- `const Plane & getPlaneOfExistence_derived (void) const`

Private Attributes

- `const Plane * plane_of_existence_m`

The documentation for this class was generated from the following file:

- `referred_to_plane_of_existence.h`

6.807 f_bytebuffer_class::resetbufferpointer Interface Reference

Public Member Functions

- subroutine `bytebuffer_class_resetbufferpointer (this)`

The documentation for this interface was generated from the following file:

- `f_bytebuffer_class.f90`

6.808 f_dividedpolygon_class::resetcentroid Interface Reference

Public Member Functions

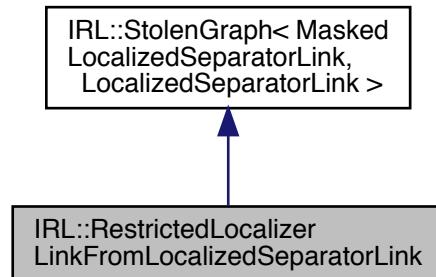
- subroutine **dividedpolygon_class_resetcentroid** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.809 IRL::RestrictedLocalizerLinkFromLocalizedSeparatorLink Class Reference

Inheritance diagram for IRL::RestrictedLocalizerLinkFromLocalizedSeparatorLink:



Collaboration diagram for IRL::RestrictedLocalizerLinkFromLocalizedSeparatorLink:



Public Member Functions

- **RestrictedLocalizerLinkFromLocalizedSeparatorLink** (void)=delete
Default constructor.
- **RestrictedLocalizerLinkFromLocalizedSeparatorLink** (const [LocalizedSeparatorLink](#) *[a_localized_separator_link](#), const [WhiteListType](#) *[a_white_list](#))
Construct with pointer to LocalizedSeparatorLink.
- **bool hasNeighbor** (const [UnsignedIndex_t](#) [a_neighbor_index](#)) const
- **RestrictedLocalizerLinkFromLocalizedSeparatorLink** **getNeighbor** (const [UnsignedIndex_t](#) [a_neighbor_index](#)) const
- const [PlanarLocalizer](#) & **getCurrentReconstruction** (void) const
- const [LocalizedSeparatorLink](#) * **getLinkingReconstructionAddress** (void) const
Return address of the thing that is linked.
- **~RestrictedLocalizerLinkFromLocalizedSeparatorLink** (void)=default
Default destructor.

Static Public Member Functions

- static constexpr [NullReconstruction](#) **getNextReconstruction** (void)

Private Types

- using **WhiteListType** = std::unordered_map< const [LocalizedSeparatorLink](#) *, bool >

Private Attributes

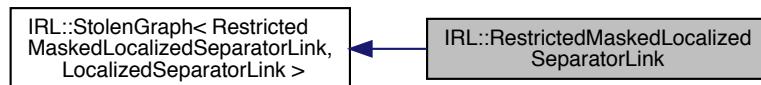
- const [WhiteListType](#) * **white_list_m**

The documentation for this class was generated from the following file:

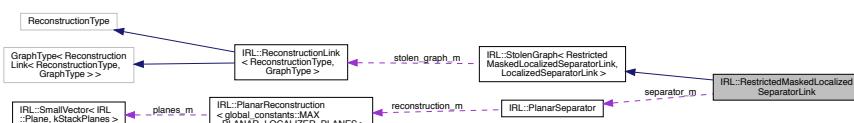
- [restricted_localizer_link_from_localized_separator_link.h](#)

6.810 IRL::RestrictedMaskedLocalizedSeparatorLink Class Reference

Inheritance diagram for IRL::RestrictedMaskedLocalizedSeparatorLink:



Collaboration diagram for IRL::RestrictedMaskedLocalizedSeparatorLink:



Public Member Functions

- [RestrictedMaskedLocalizedSeparatorLink](#) (void)=delete
Default constructor.
- [RestrictedMaskedLocalizedSeparatorLink](#) (const [LocalizedSeparatorLink](#) **a_localized_separator_link*, const [PlanarSeparator](#) **a_separator*, const [WhiteListType](#) **a_white_list*)
Construct with pointer to LocalizedSeparatorLink.
- bool [hasNeighbor](#) (const UnsignedIndex_t *a_neighbor_index*) const
- [RestrictedMaskedLocalizedSeparatorLink](#) **getNeighbor** (const UnsignedIndex_t *a_neighbor_index*) const
- const [PlanarLocalizer](#) & [getCurrentReconstruction](#) (void) const
- const [PlanarSeparator](#) & [getNextReconstruction](#) (void) const
- const [LocalizedSeparatorLink](#) * [getLinkingReconstructionAddress](#) (void) const
Return address of the thing that is linked.
- [~RestrictedMaskedLocalizedSeparatorLink](#) (void)=default
Default destructor.

Private Types

- using **WhiteListType** = std::unordered_map< const [LocalizedSeparatorLink](#) *, bool >

Private Attributes

- const [PlanarSeparator](#) * **separator_m**
- const WhiteListType * **white_list_m**

The documentation for this class was generated from the following file:

- [restricted_masked_localized_separator_link.h](#)

6.811 f_tri_class::reverseptordering Interface Reference

Public Member Functions

- subroutine **tri_class_reverseptordering** (this)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.812 f_dividedpolygon_class::reverseptordering Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_reverseptordering** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.813 f_polygon_class::reverseptordering Interface Reference

Public Member Functions

- subroutine **polygon_class_reverseptordering** (this)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.814 IRL::Secant< OptimizingClass > Class Template Reference

A templated class driver that performs the [Secant](#) method for optimization.

```
#include <secant.h>
```

Public Member Functions

- [Secant](#) (void)

Default constructor.
- void [solve](#) (OptimizingClass *a_setup_otype, const double a_initial_delta)

Set otype_m to a new pointer and then call [solve\(\)](#).
- void [solveAndTrackBounds](#) (OptimizingClass *a_setup_otype, const double a_initial_delta)

Set pointer for otype_m and then call [solveAndTrackBounds\(\)](#)
- int [getReason](#) (void) const

Return reason for exiting by integer.
- [~Secant](#) (void)=default

Default destructor.

Private Member Functions

- void [solve](#) (const double a_initial_delta)

Perform optimization.
- void [solveAndTrackBounds](#) (const double a_initial_delta)

Perform optimization and track best answers on either side of the zero to finish by bisection if necessary.

Private Attributes

- OptimizingClass * [otype_m](#)

Pointer to object of class OptimizingClass that is being optimized.
- double [delta_m](#)

Change in parameter being optimized.
- UnsignedIndex_t [iteration_m](#)

Iterations of the Secant method.
- int [reason_for_exit_m](#)

Integer indicating reason for Secant method exiting.

6.814.1 Detailed Description

```
template<class OptimizingClass>
class IRL::Secant< OptimizingClass >
```

A templated class driver that performs the [Secant](#) method for optimization.

This class is a general Gauss Newton solver that takes a pointer to a templated class that must implement certain methods (detailed below). Upon completion, the final solution should be stored in the pointed-to-class because of the use of `updateGuess()`.

Requirements for Optimizing class: `calculateSignedScalarError()` : A method to calculate a scalar error for the class with sign of (correct-guess).

- `updateGuess(...)` : A method that takes in the delta change and computes a new guess vector (which it is storing itself)
- `updateBoundaries(...)` : (Only if [solveAndTrackBounds\(\)](#) is being used) A method to keep track of the best current solutions on the +/- side of the zero.
- `errorTooHigh(...)` : A method that takes error and returns a boolean whether the error is low enough to stop optimization and return. `-iterationTooHigh(...)` : A method that takes the number of iterations and returns a bool whether the maximum number of allowable iterations has been exceeded.

6.814.2 Member Function Documentation

6.814.2.1 `solve()` [1/2]

```
template<class OptimizingClass >
void IRL::Secant< OptimizingClass >::solve (
    OptimizingClass * a_setup_otype,
    const double a_initial_delta )
```

Set `otype_m` to a new pointer and then call [solve\(\)](#).

Parameters

in	<code>a_setup_otype</code>	Pointer to object that has been setup and will be optimized.
in	<code>a_initial_delta</code>	Initial step to be taken.

6.814.2.2 `solve()` [2/2]

```
template<class OptimizingClass >
void IRL::Secant< OptimizingClass >::solve (
    const double a_initial_delta ) [private]
```

Perform optimization.

Parameters

in	<i>a_initial_delta</i>	Initial step to be taken.
----	------------------------	---------------------------

6.814.2.3 solveAndTrackBounds()

```
template<class OptimizingClass >
void IRL::Secant< OptimizingClass >::solveAndTrackBounds (
    const double a_initial_delta ) [private]
```

Perform optimization and track best answers on either side of the zero to finish by bisection if necessary.

This requires the `updateBoundaries()` method mentioned in the class description.

6.814.3 Member Data Documentation**6.814.3.1 reason_for_exit_m**

```
template<class OptimizingClass >
int IRL::Secant< OptimizingClass >::reason_for_exit_m [private]
```

Integer indicating reason for [Secant](#) method exiting.

Reasons:

- ≥ 0 : Number of iterations taken reduce error to acceptable level.
- -1 : Exited due to exceeding maximum number of iterations.

The documentation for this class was generated from the following file:

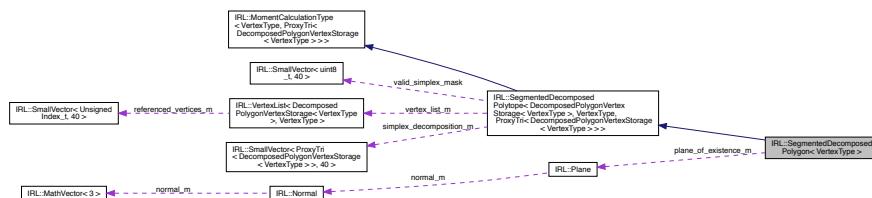
- `secant.h`

6.815 IRL::SegmentedDecomposedPolygon< VertexType > Class Template Reference

Inheritance diagram for IRL::SegmentedDecomposedPolygon< VertexType >:



Collaboration diagram for IRL::SegmentedDecomposedPolygon< VertexType >:



Public Types

- using **VertexStorageType** = `DecomposedPolygonVertexStorage< VertexType >`
- using **ProxyType** = `ProxyTri< DecomposedPolygonVertexStorage< VertexType > >>`
- using **BaseClass** = `SegmentedDecomposedPolytope< DecomposedPolygonVertexStorage< VertexType >, VertexType, ProxyTri< DecomposedPolygonVertexStorage< VertexType > >>>`

Public Member Functions

- template<class GeometryType > **SegmentedDecomposedPolygon** (const GeometryType &a_geometry, **VertexStorageType** &a_vertex_← storage)
- **SegmentedDecomposedPolygon** (**VertexStorageType** &a_vertex_storage)
- **ProxyType** **getNullProxyObject** (void) const
- void **setNumberOfSimplicesInDecomposition** (const UnsignedIndex_t a_new_size)
- void **push_back** (const **ProxyType** &a_proxy)
- void **setPlaneOfExistence** (const **Plane** *a_plane)
- const **Plane** & **getPlaneOfExistence** (void) const
- **VolumeMomentsAndNormal** **calculateVolumeMomentsAndNormal** (void) const

Private Attributes

- const **Plane** * **plane_of_existence_m**

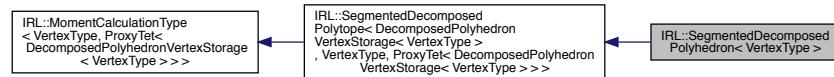
Additional Inherited Members

The documentation for this class was generated from the following file:

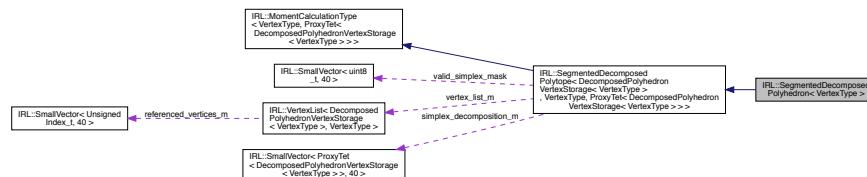
- `segmented_decomposed_polytope.h`

6.816 IRL::SegmentedDecomposedPolyhedron< VertexType > Class Template Reference

Inheritance diagram for IRL::SegmentedDecomposedPolyhedron< VertexType >:



Collaboration diagram for IRL::SegmentedDecomposedPolyhedron< VertexType >:



Public Types

- using **VertexStorageType** = `DecomposedPolyhedronVertexStorage< VertexType >`
- using **ProxyType** = `ProxyTet< DecomposedPolyhedronVertexStorage< VertexType >>>`
- using **BaseClass** = `SegmentedDecomposedPolytope< DecomposedPolyhedronVertexStorage< VertexType >, VertexType, ProxyTet< DecomposedPolyhedronVertexStorage< VertexType >>>`

Public Member Functions

- template<class GeometryType> **SegmentedDecomposedPolyhedron** (const GeometryType &a_geometry, **VertexStorageType** &a_vertex_storage)
- **SegmentedDecomposedPolyhedron** (**VertexStorageType** &a_vertex_storage)
- void **push_back** (const **ProxyType** &a_proxy)

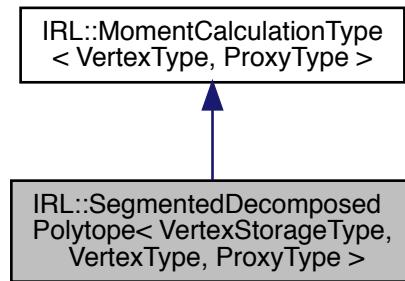
Additional Inherited Members

The documentation for this class was generated from the following file:

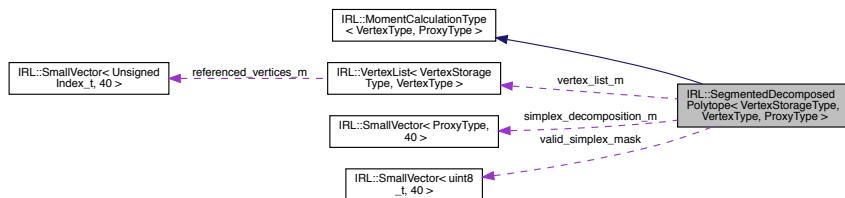
- `segmented_decomposed_polytope.h`

6.817 IRL::SegmentedDecomposedPolytope< VertexStorageType, VertexType, ProxyType > Class Template Reference

Inheritance diagram for IRL::SegmentedDecomposedPolytope< VertexStorageType, VertexType, ProxyType >:



Collaboration diagram for IRL::SegmentedDecomposedPolytope< VertexStorageType, VertexType, ProxyType >:



Public Types

- using **pt_type** = VertexType

Public Member Functions

- **SegmentedDecomposedPolytope** (VertexStorageType &a_vertex_storage)
- **VertexList< VertexStorageType, VertexType > & getVertexList** (void)
- const **VertexList< VertexStorageType, VertexType > & getVertexList** (void) const
- UnsignedIndex_t **getNumberOfSimplicesInDecomposition** (void) const
- ProxyType **getNullProxyObject** (void) const
- void **setNumberOfSimplicesInDecomposition** (const UnsignedIndex_t a_new_size)
- ProxyType & **getSimplexFromDecomposition** (const UnsignedIndex_t a_tet)
- const ProxyType & **getSimplexFromDecomposition** (const UnsignedIndex_t a_tet) const
- void **setNumberOfValidSimplices** (const UnsignedIndex_t a_new_size)
- void **markAsValid** (const UnsignedIndex_t a_index)
- void **markAsNotValid** (const UnsignedIndex_t a_index)
- bool **isValid** (const UnsignedIndex_t a_index) const
- bool **isNotValid** (const UnsignedIndex_t a_index) const
- void **clear** (void)

Protected Attributes

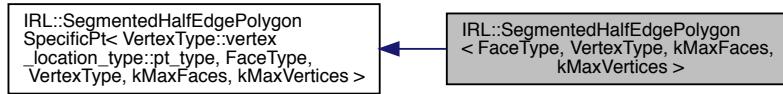
- `SmallVector< ProxyType, 40 > simplex_decomposition_m`
- `SmallVector< uint8_t, 40 > valid_simplex_mask`
- `VertexList< VertexStorageType, VertexType > vertex_list_m`

The documentation for this class was generated from the following file:

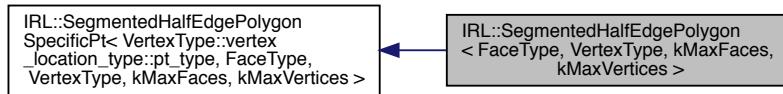
- `segmented_decomposed_polytope.h`

6.818 IRL::SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference

Inheritance diagram for IRL::SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices >:



Collaboration diagram for IRL::SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices >:



The documentation for this class was generated from the following file:

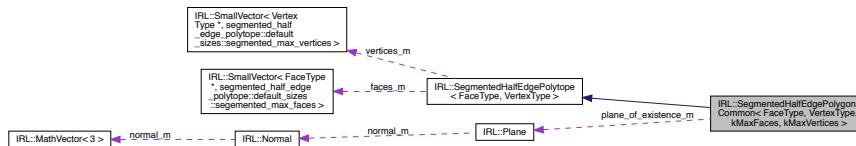
- `segmented_half_edge_polygon.h`

6.819 IRL::SegmentedHalfEdgePolygonCommon< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference

Inheritance diagram for IRL::SegmentedHalfEdgePolygonCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >:



Collaboration diagram for IRL::SegmentedHalfEdgePolygonCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >:



Public Member Functions

- void **setPlaneOfExistence** (const **Plane** **a_plane*)
- const **Plane** & **getPlaneOfExistence** (void) const
- **Volume** **calculateVolume** (void) const
- **VolumeMomentsAndNormal** **calculateVolumeMomentsAndNormal** (void) const
- **Pt** **calculateCentroid** (void) const
- **VolumeMoments** **calculateMoments** (void) const
- bool **checkValidHalfEdgeStructure** (void)

Private Member Functions

- bool **checkVerticesCompleteCycle** (void) const
- bool **checkVerticesOnlyHaveHalfEdgesWithCorrectFaces** (void) const

Private Attributes

- const **Plane** * **plane_of_existence_m**

Additional Inherited Members

The documentation for this class was generated from the following file:

- segmented_half_edge_polygon.h

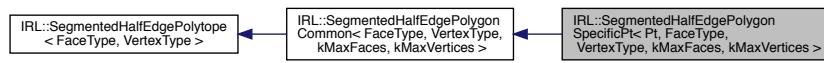
6.820 IRL::SegmentedHalfEdgePolygonSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference

The documentation for this class was generated from the following file:

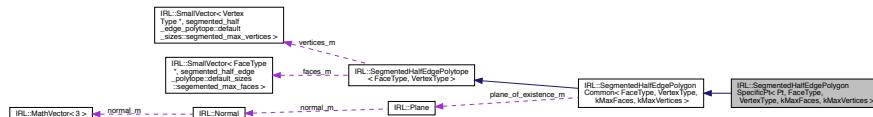
- `segmented_half_edge_polygon.h`

6.821 IRL::SegmentedHalfEdgePolygonSpecificPt< Pt, FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference

Inheritance diagram for IRL::SegmentedHalfEdgePolygonSpecificPt< Pt, FaceType, VertexType, kMaxFaces, kMaxVertices >:



Collaboration diagram for IRL::SegmentedHalfEdgePolygonSpecificPt< Pt, FaceType, VertexType, kMaxFaces, kMaxVertices >:



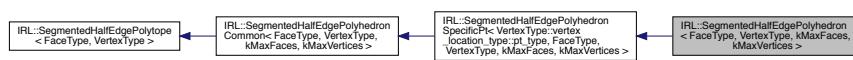
Additional Inherited Members

The documentation for this class was generated from the following file:

- `segmented_half_edge_polygon.h`

6.822 IRL::SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference

Inheritance diagram for IRL::SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices >:



Collaboration diagram for IRL::SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices >:



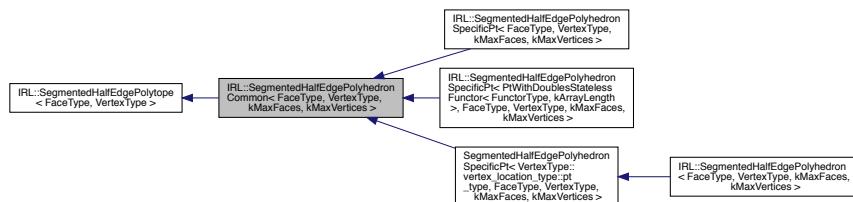
Additional Inherited Members

The documentation for this class was generated from the following file:

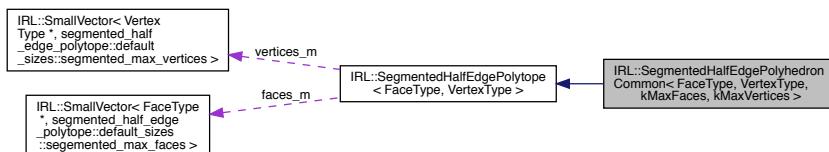
- segmented_half_edge_polyhedron.h

6.823 IRL::SegmentedHalfEdgePolyhedronCommon< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference

Inheritance diagram for IRL::SegmentedHalfEdgePolyhedronCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >:



Collaboration diagram for IRL::SegmentedHalfEdgePolyhedronCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >:



Public Member Functions

- [Volume calculateVolume \(void\)](#)
- [Pt calculateCentroid \(void\)](#)
- [VolumeMoments calculateMoments \(void\)](#)
- [bool checkValidHalfEdgeStructure \(void\)](#)

Private Member Functions

- void **markFacesTouchedByOpposite** (FaceType *a_face_to_touch_from, std::unordered_map< FaceType *, bool > *a_map_to_fill)
- bool **checkVerticesCompleteCycle** (void) const
- bool **checkVerticesOnlyHaveHalfEdgesWithCorrectFaces** (void) const

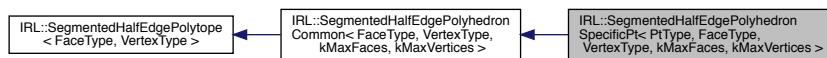
Additional Inherited Members

The documentation for this class was generated from the following file:

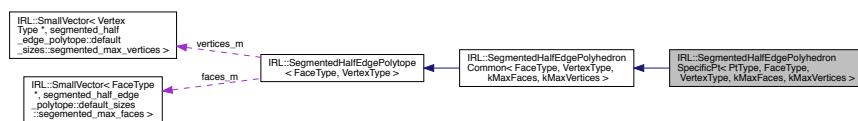
- segmented_half_edge_polyhedron.h

6.824 IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference

Inheritance diagram for IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices >:



Collaboration diagram for IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices >:



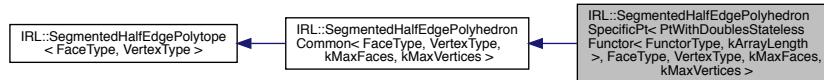
Additional Inherited Members

The documentation for this class was generated from the following file:

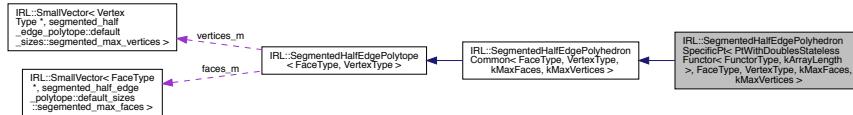
- segmented_half_edge_polyhedron.h

6.825 IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference

Inheritance diagram for IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, FaceType, VertexType, kMaxFaces, kMaxVertices >:



Collaboration diagram for IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, FaceType, VertexType, kMaxFaces, kMaxVertices >:



Public Member Functions

- [VolumeMomentsAndDoubles< kArrayLength > calculateVolumeMomentsAndDoubles \(void\)](#)

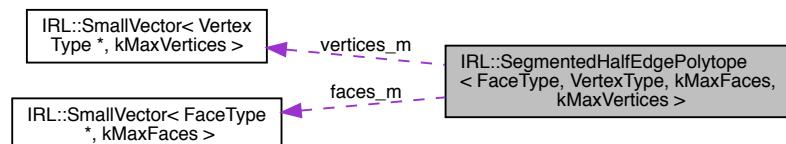
Additional Inherited Members

The documentation for this class was generated from the following file:

- segmented_half_edge_polyhedron.h

6.826 IRL::SegmentedHalfEdgePolytope< FaceType, VertexType, kMaxFaces, kMaxVertices > Class Template Reference

Collaboration diagram for IRL::SegmentedHalfEdgePolytope< FaceType, VertexType, kMaxFaces, kMaxVertices >:



Public Types

- using **face_type** = FaceType
- using **vertex_type** = VertexType
- using **vertex_location_type** = typename vertex_type::vertex_location_type
- using **pt_type** = typename vertex_location_type::pt_type

Public Member Functions

- void **setNumberOfFaces** (const UnsignedIndex_t a_number_of_faces)
- void **setNumberOfVertices** (const UnsignedIndex_t a_size)
- UnsignedIndex_t **getNumberOfFaces** (void) const
- UnsignedIndex_t **getNumberOfVertices** (void) const
- FaceType * **operator[]** (const UnsignedIndex_t a_index)
- const FaceType * **operator[]** (const UnsignedIndex_t a_index) const
- void **addFace** (FaceType *a_face)
- void **addVertex** (VertexType *a_vertex)
- VertexType * **getVertex** (const UnsignedIndex_t a_index)
- const VertexType * **getVertex** (const UnsignedIndex_t a_index) const
- VertexType *& **getVertexPointer** (const UnsignedIndex_t a_index)
- int **calculateAndStoreDistanceToVertices** (const Plane &a_plane)
- void **removeFace** (const UnsignedIndex_t a_index)
- void **removeVertex** (const UnsignedIndex_t a_index)
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept

Static Public Attributes

- static constexpr UnsignedIndex_t **maxVertices** = kMaxVertices
- static constexpr UnsignedIndex_t **maxFaces** = kMaxFaces

Private Types

- using **iterator** = typename SmallVector< FaceType *, kMaxFaces >::iterator
- using **const_iterator** = typename SmallVector< FaceType *, kMaxFaces >::const_iterator

Private Member Functions

- void **checkIfStaticAllocationExceeded** (void) const

Private Attributes

- **SmallVector<** FaceType *, kMaxFaces **> faces_m**
- **SmallVector<** VertexType *, kMaxVertices **> vertices_m**

The documentation for this class was generated from the following file:

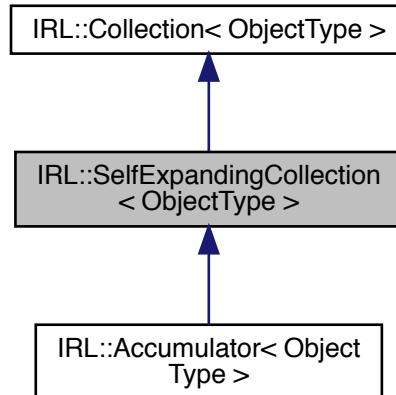
- segmented_half_edge_polytope.h

6.827 IRL::SelfExpandingCollection< ObjectType > Class Template Reference

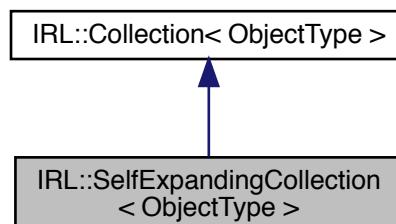
A modified [Collection](#) class that self expands when using the operator[] would force out of memory access.

```
#include <self_expanding_collection.h>
```

Inheritance diagram for IRL::SelfExpandingCollection< ObjectType >:



Collaboration diagram for IRL::SelfExpandingCollection< ObjectType >:



Public Member Functions

- [`SelfExpandingCollection`](#) (`void`)=`default`
Default constructor.
- `ObjectType & operator[]` (`const UnsignedIndex_t a_index`)
This will self-expand to prevent itself from accessing out of bounds memory.
- `const ObjectType & operator[]` (`const UnsignedIndex_t a_index`) `const`
Const version for access to object in collection.
- [`~SelfExpandingCollection`](#) (`void`)=`default`
Default destructor.

Additional Inherited Members

6.827.1 Detailed Description

```
template<class ObjectType>
class IRL::SelfExpandingCollection< ObjectType >
```

A modified [Collection](#) class that self expands when using the operator[] would force out of memory access.

The documentation for this class was generated from the following file:

- `self_expanding_collection.h`

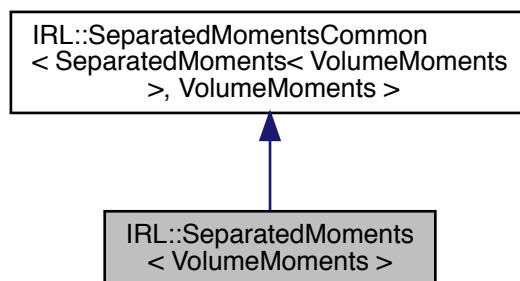
6.828 IRL::SeparatedMoments< MomentsType > Class Template Reference

The documentation for this class was generated from the following file:

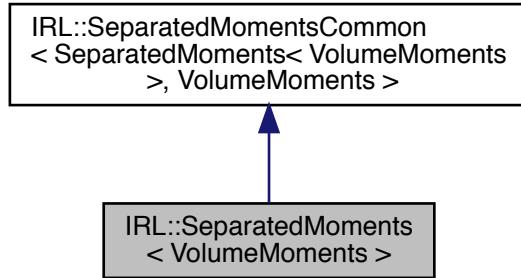
- `separated_volume_moments.h`

6.829 IRL::SeparatedMoments< VolumeMoments > Class Template Reference

Inheritance diagram for IRL::SeparatedMoments< VolumeMoments >:



Collaboration diagram for IRL::SeparatedMoments< VolumeMoments >:



Static Public Member Functions

- template<class GeometryType>
static **SeparatedMoments** **fillWithComplementMoments** (const **VolumeMoments** &a_known_moments, const GeometryType &a_encompassing_geometry, const bool a_flipped)
- static **SeparatedMoments** **fillWithComplementMoments** (const **VolumeMoments** &a_known_moments, const **VolumeMoments** &a_encompassing_geometry_volume_moments, const bool a_flipped)

Private Types

- using **SelfType** = **SeparatedMoments< VolumeMoments >**
- using **MomentsType** = **VolumeMoments**

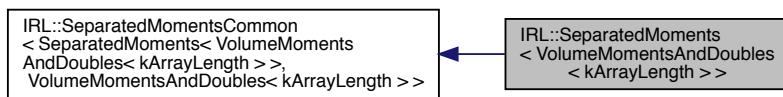
Additional Inherited Members

The documentation for this class was generated from the following file:

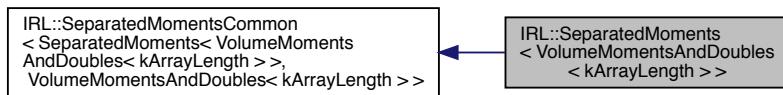
- separated_volume_moments.h

6.830 IRL::SeparatedMoments< VolumeMomentsAndDoubles< kArrayLength > > Class Template Reference

Inheritance diagram for IRL::SeparatedMoments< VolumeMomentsAndDoubles< kArrayLength > >:



Collaboration diagram for IRL::SeparatedMoments< VolumeMomentsAndDoubles< kArrayLength > >:



Static Public Member Functions

- template<class GeometryType>
static **SeparatedMoments** **fillWithComplementMoments** (const **VolumeMomentsAndDoubles**< kArrayLength > &a_known_moments, const GeometryType &a_encompassing_geometry, const bool a_flipped)
- static **SeparatedMoments** **fillWithComplementMoments** (const **VolumeMomentsAndDoubles**< kArrayLength > &a_known_moments, const **VolumeMomentsAndDoubles**< kArrayLength > &a_encompassing_geometry_volume_moments, const bool a_flipped)

Private Types

- using **SelfType** = **SeparatedMoments**< **VolumeMomentsAndDoubles**< kArrayLength > >
- using **MomentsType** = **VolumeMomentsAndDoubles**< kArrayLength >

Additional Inherited Members

The documentation for this class was generated from the following file:

- separated_volume_moments.h

6.831 IRL::SeparatedMomentsCommon< Derived, MomentsType > Class Template Reference

Storage for multiple volume moments.

```
#include <separated_volume_moments.h>
```

Public Types

- using **moments_type** = **MomentsType**

Public Member Functions

- `SeparatedMomentsCommon (void)=default`
Default constructor.
- `constexpr SeparatedMomentsCommon (const MomentsType &a_moment_0, const MomentsType &a_moment_1)`
Constructor that sets moments for two phases.
- `MomentsType & operator[] (const UnsignedIndex_t a_moment_index)`
- `const MomentsType & operator[] (const UnsignedIndex_t a_moment_index) const`
- `Derived & operator+= (const Derived &a_rhs)`
Overload += operator to adjust liquid and gas moments.
- `Derived & operator*=(const double a_rhs)`
*Overload *= operator to for const doubles.*
- `Derived & operator=(const double a_value)`
Overload assignment to assign constant value to moments.
- `void normalizeByVolume (void)`
Normalize both centroids by their respective phase volumes.
- `void multiplyByVolume (void)`
Multiply both centroids by their respective phase volumes.
- `void swap (const UnsignedIndex_t a_index_0, const UnsignedIndex_t a_index_1)`
Swap two stores moments.
- `iterator begin (void) noexcept`
- `const_iterator begin (void) const noexcept`
- `const_iterator cbegin (void) const noexcept`
- `iterator end (void) noexcept`
- `const_iterator end (void) const noexcept`
- `const_iterator cend (void) const noexcept`
- `~SeparatedMomentsCommon (void)=default`
Default destructor.

Static Public Member Functions

- `static Derived fromRawDoublePointer (const double *a_list)`
- `static Derived fromScalarConstant (const double a_value)`
- `template<class GeometryType > static Derived fillWithComplementMoments (const MomentsType &a_known_moments, const GeometryType &a_encompassing_geometry, const bool a_flipped)`
Use knowledge of the initial geometry and moments of the cut phase to calculate the other phase's moments, return VOLUME WEIGHTED SeparatedMomentsCommon.
- `static Derived fillWithComplementMoments (const MomentsType &a_known_moments, const MomentsType &a_encompassing_geometry_volume_moments, const bool a_flipped)`
- `static constexpr UnsignedIndex_t getNumberOfPhases (void)`

Private Types

- `using iterator = typename std::array< MomentsType, 2 >::iterator`
- `using const_iterator = typename std::array< MomentsType, 2 >::const_iterator`

Private Member Functions

- Derived & **getDerived** (void)
- const Derived & **getDerived** (void) const
- constexpr **SeparatedMomentsCommon** (const double *a_list)
Set volume moments from a list of 8 doubles.
- **SeparatedMomentsCommon** (const double a_value)
Construct that initializes volume/centroid as a value.

Private Attributes

- std::array< MomentsType, kMaxNumberOfPhases > **volume_moments_m**
VolumeMoments.

Static Private Attributes

- static constexpr UnsignedIndex_t **kMaxNumberOfPhases** = 2

6.831.1 Detailed Description

```
template<class Derived, class MomentsType>
class IRL::SeparatedMomentsCommon< Derived, MomentsType >
```

Storage for multiple volume moments.

6.831.2 Constructor & Destructor Documentation

6.831.2.1 SeparatedMomentsCommon()

```
template<class Derived, class MomentsType>
constexpr IRL::SeparatedMomentsCommon< Derived, MomentsType >::SeparatedMomentsCommon (
    const double * a_list ) [explicit], [private]
```

Set volume moments from a list of 8 doubles.

Construct the volume moments from a list of 8 doubles. Expected order is InternalVolumeMoments(**Volume**,Centroidx_—, Centroid_y, Centroid_z), then ExternalVolumeMoments(**Volume**,Centroidx_—, Centroid_y, Centroid_z).

6.831.3 Member Function Documentation

6.831.3.1 fillWithComplementMoments()

```
template<class Derived, class MomentsType>
template<class GeometryType >
static Derived IRL::SeparatedMomentsCommon< Derived, MomentsType >::fillWithComplementMoments
(
    const MomentsType & a_known_moments,
    const GeometryType & a_encompassing_geometry,
    const bool a_flipped ) [inline], [static]
```

Use knowledge of the initial geometry and moments of the cut phase to calculate the other phase's moments, return VOLUME WEIGHTED [SeparatedMomentsCommon](#).

This function uses knowledge of the initial geometry (such as a rectangular cuboid or tet) and the moments known from the computational cutting to obtain the moments for the other phase.

NOTICE: The moments in `a_known_moments` are actually volume and volume*centroid.

ASIDE: These moments are first assumed to be the liquid phase in the [SeparatedMomentsCommon](#) object that is returned. The bool `a_flipped` is then used to store in [SeparatedMomentsCommon](#) as `{a_known_moments, unknown_moments}` (if true) or `{unknown_moments, a_known_moments}` if false.

Template Requirements for `GeometryType`:

- A `volume(void)` method that returns the volume of the encompassing geometry
- A `centroid(void)` method that returns the centroid (NOT volume*centroid) of the encompassing geometry

Parameters

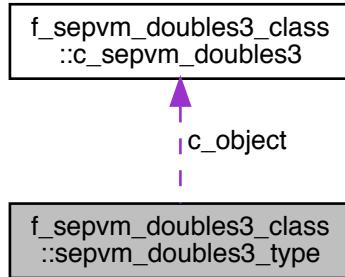
in	<code>a_known_moments</code>	GeometricMoments that are known.
in	<code>a_encompassing_geometry</code>	The geometry from which the known moments are a subset.
in	<code>a_flipped</code>	Boolean expressing whether the supplied <code>a_known_moments</code> is for SeparatedMomentsCommon[0] or <code>[1]</code> .

The documentation for this class was generated from the following file:

- `separated_volume_moments.h`

6.832 f_sepvm_doubles3_class::sepvm_doubles3_type Type Reference

Collaboration diagram for f_sepvm_doubles3_class::sepvm_doubles3_type:



Public Member Functions

- final **sepvm_doubles3_class_delete**

Public Attributes

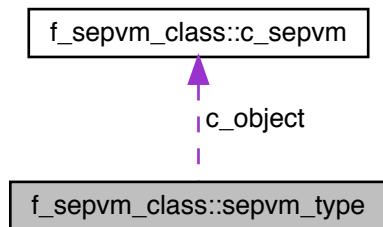
- type([c_sepvm_doubles3](#)) **c_object**

The documentation for this type was generated from the following file:

- [f_sepvm_doubles3_class.f90](#)

6.833 f_sepvm_class::sepvm_type Type Reference

Collaboration diagram for f_sepvm_class::sepvm_type:



Public Member Functions

- final **sepvm_class_delete**

Private Attributes

- type([c_sepvm](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_sepvm_class.f90](#)

6.834 f_serializer::serializeandpack Interface Reference

Public Member Functions

- subroutine **serializeandpack_planarseparator_bytebuffer** (a_separator, a_byte_buffer)

The documentation for this interface was generated from the following file:

- [f_serializer.f90](#)

6.835 f_lviraneighborhood_rectangularcuboid_class::setcenterofstencil Interface Reference

Public Member Functions

- subroutine **lviraneighborhood_rectangularcuboid_class_setcenterofstencil** (this, a_center_cell_index)

The documentation for this interface was generated from the following file:

- [f_lviraneighborhood_rectangularcuboid_class.f90](#)

6.836 f_r2pneighborhood_rectangularcuboid_class::setcenterofstencil Interface Reference

Public Member Functions

- subroutine **r2pneighborhood_rectangularcuboid_class_setcenterofstencil** (this, a_center_cell_index)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.837 f_polyhedron24_doubles3_class::setdata Interface Reference

Public Member Functions

- subroutine **polyhedron24_doubles3_class_setdata** (this, a_index, a_data)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.838 f_cappeddodecahedron_doubles3_class::setdata Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_doubles3_class_setdata** (this, a_index, a_data)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.839 f_volumefractionmatching::setdistancetomatchvolumefraction Interface Reference

Public Member Functions

- subroutine **setdistancetomatchvolumefraction_rc_ps** (a_rectangular_cuboid, a_volume_fraction, a_← planar_separator, a_volume_fraction_tolerance)
- subroutine **setdistancetomatchvolumefraction_rc_ps_deftol** (a_rectangular_cuboid, a_volume_fraction, a_planar_separator)

The documentation for this interface was generated from the following file:

- [f_volumefractionmatching.f90](#)

6.840 f_localizedseparatorlink_class::setedgeconnectivity Interface Reference

Public Member Functions

- subroutine **localizedseparatorlink_class_setedgeconnectivity** (this, a_plane_index, a_neighboring_← LocalizedSeparatorLink)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.841 f_localizerlink_class::setedgeconnectivity Interface Reference

Public Member Functions

- subroutine **localizerlink_class_setedgeconnectivity** (this, a_plane_index, a_neighboring_LocalizerLink)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.842 f_localizedseparatorlink_class::setedgeconnectivitynull Interface Reference

Public Member Functions

- subroutine **localizedseparatorlink_class_setedgeconnectivitynull** (this, a_plane_index)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.843 f_localizerlink_class::setedgeconnectivitynull Interface Reference

Public Member Functions

- subroutine **localizerlink_class_setedgeconnectivitynull** (this, a_plane_index)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.844 f_planarlocalizer_class::setfromrectangularcuboid Interface Reference

Public Member Functions

- subroutine **planarlocalizer_class_setfromrectangularcuboid** (this, a_lower_pt, a_upper_pt)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.845 f_localizerlink_class::setid Interface Reference

Public Member Functions

- subroutine **localizerlink_class_setid** (this, a_id)

The documentation for this interface was generated from the following file:

- [f_localizerlink_class.f90](#)

6.846 f_localizedseparatorlink_class::setid Interface Reference

Public Member Functions

- subroutine **localizedseparatorlink_class_setid** (this, a_id)

The documentation for this interface was generated from the following file:

- [f_localizedseparatorlink_class.f90](#)

6.847 f_elviraneighborhood_class::setmember Interface Reference

Public Member Functions

- subroutine **elviraneighborhood_class_setmember** (this, a_rectangular_cuboid, a_liquid_volume_fraction, i, j, k)

The documentation for this interface was generated from the following file:

- [f_elviraneighborhood_class.f90](#)

6.848 f_r2pneighborhood_rectangularcuboid_class::setmember Interface Reference

Public Member Functions

- subroutine **r2pneighborhood_rectangularcuboid_class_setmember** (this, a_rectangular_cuboid, a_separated_volume_moments, a_index)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.849 f_lviraneighborhood_rectangularcuboid_class::setmember Interface Reference

Public Member Functions

- subroutine **lviraneighborhood_rectangularcuboid_class_setmember** (this, a_index, a_rectangular_cuboid, a_liquid_volume_fraction)

The documentation for this interface was generated from the following file:

- [f_lviraneighborhood_rectangularcuboid_class.f90](#)

6.850 f_planarseparator_class::setnumberofplanes Interface Reference

Public Member Functions

- subroutine **planarseparator_class_setnumberofplanes** (this, a_number_to_set)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.851 f_planarlocalizer_class::setnumberofplanes Interface Reference

Public Member Functions

- subroutine **planarlocalizer_class_setnumberofplanes** (this, a_number_to_set)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.852 f_planarlocalizer_class::setplane Interface Reference

Public Member Functions

- subroutine **planarlocalizer_class_setplane** (this, a_plane_index_to_set, a_normal, a_distance)

The documentation for this interface was generated from the following file:

- [f_planarlocalizer_class.f90](#)

6.853 f_planarseparator_class::setplane Interface Reference

Public Member Functions

- subroutine **planarseparator_class_setplane** (this, a_plane_index_to_set, a_normal, a_distance)

The documentation for this interface was generated from the following file:

- [f_planarseparator_class.f90](#)

6.854 f_tri_class::setplaneofexistence Interface Reference

Public Member Functions

- subroutine **tri_class_setplaneofexistence** (this, a_plane)

The documentation for this interface was generated from the following file:

- [f_tri_class.f90](#)

6.855 f_polygon_class::setplaneofexistence Interface Reference

Public Member Functions

- subroutine **polygon_class_setplaneofexistence** (this, a_plane)

The documentation for this interface was generated from the following file:

- [f_polygon_class.f90](#)

6.856 f_dividedpolygon_class::setplaneofexistence Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_setplaneofexistence** (this, a_plane)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.857 f_polyhedron24_doubles3_class::setpt Interface Reference

Public Member Functions

- subroutine **polyhedron24_doubles3_class_setpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_doubles3_class.f90](#)

6.858 f_cappeddodecahedron_doubles3_class::setpt Interface Reference

Public Member Functions

- subroutine **cappeddodecahedron_doubles3_class_setpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_cappeddodecahedron_doubles3_class.f90](#)

6.859 f_polyhedron24_class::setpt Interface Reference

Public Member Functions

- subroutine **polyhedron24_class_setpt** (this, a_index, a_pt)

The documentation for this interface was generated from the following file:

- [f_polyhedron24_class.f90](#)

6.860 f_elviraneighborhood_class::setszie Interface Reference

Public Member Functions

- subroutine **elviraneighborhood_class_setszie** (this, a_size)

The documentation for this interface was generated from the following file:

- [f_elviraneighborhood_class.f90](#)

6.861 f_lviraneighborhood_rectangularcuboid_class::setszie Interface Reference

Public Member Functions

- subroutine **lviraneighborhood_rectangularcuboid_class_setszie** (this, a_size)

The documentation for this interface was generated from the following file:

- [f_lviraneighborhood_rectangularcuboid_class.f90](#)

6.862 f_r2pneighborhood_rectangularcuboid_class::setszie Interface Reference

Public Member Functions

- subroutine **r2pneighborhood_rectangularcuboid_class_setszie** (this, a_size)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.863 f_bytebuffer_class::setszie Interface Reference

Public Member Functions

- subroutine **bytebuffer_class_setszie** (this, a_size)

The documentation for this interface was generated from the following file:

- [f_bytebuffer_class.f90](#)

6.864 f_r2pneighborhood_rectangularcuboid_class::setsurfacearea Interface Reference

Public Member Functions

- subroutine **r2pneighborhood_rectangularcuboid_class_setsurfacearea** (this, a_surface_area)

The documentation for this interface was generated from the following file:

- [f_r2pneighborhood_rectangularcuboid_class.f90](#)

6.865 IRL::ShareVolumeAbovePlaneWithLinkedNeighbor Struct Reference

Static Public Member Functions

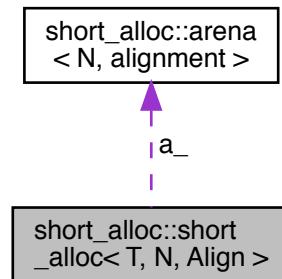
- template<class SimplexType , class ReconstructionType , class VerticesList , class ReturnType >
static void generateAndHandleSimplicesFromVolumeAbovePlane (const ReconstructionType &a_<reconstruction, const LookupIndex_t a_cutting_case_for_simplex_and_current_plane, const VerticesList &a_simplex_vertices_and_intersection_points, const UnsignedIndex_t a_index_for_plane_that_just_cut, ReturnType *a_moments_to_return)

The documentation for this struct was generated from the following file:

- continue_dividing_volume.h

6.866 short_alloc::short_alloc< T, N, Align > Class Template Reference

Collaboration diagram for short_alloc::short_alloc< T, N, Align >:



Classes

- struct [rebind](#)

Public Types

- using **value_type** = T
- using **arena_type** = [arena< size, alignment >](#)

Public Member Functions

- short_alloc** (const [short_alloc](#) &)=default
- short_alloc** & **operator=** (const [short_alloc](#) &)=delete
- short_alloc** ([arena_type](#) &a) noexcept
- template<class U >
short_alloc (const [short_alloc](#)< U, N, alignment > &a) noexcept
- T * **allocate** (std::size_t n)
- void **deallocate** (T *p, std::size_t n) noexcept

Static Public Attributes

- static auto constexpr **alignment** = Align
- static auto constexpr **size** = N

Private Attributes

- **arena_type** & **a_**

Friends

- template<class U , std::size_t M, std::size_t A>
class **short_alloc**
- template<class T1 , std::size_t N1, std::size_t A1, class U , std::size_t M, std::size_t A2>
bool **operator==** (const **short_alloc**< T1, N1, A1 > &x, const **short_alloc**< U, M, A2 > &y) noexcept

The documentation for this class was generated from the following file:

- short_alloc.h

6.867 IRL::SimplexCutting Struct Reference

The documentation for this struct was generated from the following file:

- default_cutting_method.h

6.868 IRL::SimplexWrapper< SimplexType, Enable > Struct Template Reference

The documentation for this struct was generated from the following file:

- simplex_wrapper.h

6.869 IRL::SimplexWrapper< SimplexType, enable_if_t< is_tet< SimplexType >::value > > Struct Template Reference

Static Public Member Functions

- static constexpr LookupIndex_t **numberOfSimplicesInVolumeBelowPlaneAfterCutting** (const LookupIndex_t a_cutting_case)
- static constexpr LookupIndex_t **numberOfSimplicesInVolumeAfterCutting** (const LookupIndex_t a_cutting_case)
- static double **minimumAmountToTrack** (void)
- static constexpr bool **isSimplexFullyBelowPlane** (const LookupIndex_t a_cutting_case)
- static constexpr bool **isSimplexFullyAbovePlane** (const LookupIndex_t a_cutting_case)
- static auto **getSimplexVerticesAndIntersectionPoints** (const SimplexType &a_tet, const std::array< double, simplex_nvert > &a_vert_distances, const LookupIndex_t a_cutting_case) -> **CutTetVertices**< typename SimplexType::pt_type >

Calculates the vertices for a tet intersected with a plane.
- static constexpr auto **simplexFromCutSimplexVertices** (const **CutTetVertices**< typename SimplexType::pt_type > &a_cut_tet_vertices, const LookupIndex_t a_cutting_case, const LookupIndex_t a_tet_number_to_get) -> **ProxyTet**< **CutTetVertices**< typename SimplexType::pt_type >>

Returns a tet from tet vertices after intersection with a cutting plane.

Static Public Attributes

- static const constexpr bool **isTet** = true
- static const constexpr bool **isTri** = false
- static const constexpr LookupIndex_t **simplex_nvert** = 4
- static const constexpr LookupIndex_t **max_cut_simplex_nvert** = 8

6.869.1 Member Function Documentation

6.869.1.1 getSimplexVerticesAndIntersectionPoints()

```
template<class SimplexType >
static auto IRL::SimplexWrapper< SimplexType, enable_if_t< is_tet< SimplexType >::value > >::getSimplexVerticesAndIntersectionPoints (
    const SimplexType & a_tet,
    const std::array< double, simplex_nvert > & a_vert_distances,
    const LookupIndex_t a_cutting_case ) -> CutTetVertices< typename SimplexType::pt_type > [inline], [static]
```

Calculates the vertices for a tet intersected with a plane.

This function takes `a_tet` and the signed distance to each tet vertex, along with the unique integer identifier for the cutting configuration `a_cutting_case` to fill in a supplied `Pt[8]` array `a_vertices`. The first 4 elements in `a_vertices` are the original tet. The next vertices are vertices that lay on the intersection points between tet edges and the intersecting plane. These vertices are found using the function `ptPlaneIntersectsLine(...)`. The total number of vertices will be `4+number_of_new_vertices_after_cut[a_cutting_case]`.

Parameters

in	<code>a_tet</code>	Tet that is being cut by plane and subdivided
in	<code>a_vert_distances</code>	Double[4] array with the signed distance of each tet vertex from the cutting plane.
in	<code>a_cutting_case</code>	Unique integer ID indicating cutting configuration.
out	<code>a_vertices</code>	Array of up to 8 points which contains original tet vertices [0:3] and vertices at the intersection between tet edges and the cutting plane.

6.869.1.2 simplexFromCutSimplexVertices()

```
template<class SimplexType >
static constexpr auto IRL::SimplexWrapper< SimplexType, enable_if_t< is_tet< SimplexType >::value > >::simplexFromCutSimplexVertices (
    const CutTetVertices< typename SimplexType::pt_type > & a_cut_tet_vertices,
    const LookupIndex_t a_cutting_case,
    const LookupIndex_t a_tet_number_to_get ) -> ProxyTet< CutTetVertices< typename SimplexType::pt_type > > [inline], [static]
```

Returns a tet from tet vertices after intersection with a cutting plane.

Given an array of `a_cut_tet_vertices` (where these vertices are the original tet vertices plus any vertices resulting from intersection with a cutting plane) the tet corresponding to `a_tet_number_to_get` in `verts_for_tets[][][]` from `lookup_tables.h` is returned.

Parameters

in	<code>a_cut_tet_vertices</code>	Array of vertices of a tet after intersection with a plane. Valid length of 1-8.
in	<code>a_cutting_case</code>	The cutting case indicating signs of original tets vertices.
in	<code>a_tet_number_to_get</code>	The tet to return from <code>verts_for_tets[][][]</code> .

The documentation for this struct was generated from the following file:

- `simplex_wrapper.h`

6.870 IRL::SimplexWrapper< SimplexType, enable_if_t< is_tri< SimplexType >>::value >> Struct Template Reference

Static Public Member Functions

- static constexpr LookupIndex_t `numberOfSimplicesInVolumeBelowPlaneAfterCutting` (const LookupIndex_t `a_cutting_case`)
- static constexpr UnsignedIndex_t `numberOfSimplicesInVolumeAfterCutting` (const LookupIndex_t `a_cutting_case`)
- static double `minimumAmountToTrack` (void)
- static constexpr bool `isSimplexFullyBelowPlane` (const LookupIndex_t `a_cutting_case`)
- static constexpr bool `isSimplexFullyAbovePlane` (const LookupIndex_t `a_cutting_case`)
- static auto `getSimplexVerticesAndIntersectionPoints` (const SimplexType &`a_tri`, const std::array< double, simplex_nvert > &`a_vert_distances`, const LookupIndex_t `a_cutting_case`) -> `CutTriangleVerticesAndPlane< typename SimplexType::pt_type >`
- static auto `simplexFromCutSimplexVertices` (const `CutTriangleVerticesAndPlane< typename SimplexType::pt_type >` &`a_cut_tri_vertices`, const LookupIndex_t `a_cutting_case`, const LookupIndex_t `a_tri_number_to_get`) -> `ProxyTri< CutTriangleVerticesAndPlane< typename SimplexType::pt_type >>`

Static Public Attributes

- static const constexpr bool `isTet` = false
- static const constexpr bool `isTri` = true
- static const constexpr LookupIndex_t `simplex_nvert` = 3
- static const constexpr LookupIndex_t `max_cut_simplex_nvert` = 5

The documentation for this struct was generated from the following file:

- `simplex_wrapper.h`

6.871 IRL::SmallVector< T, BuffSizeInElements > Class Template Reference

Public Types

- using `iterator` = typename BaseVector::iterator
- using `const_iterator` = typename BaseVector::const_iterator

Public Member Functions

- **SmallVector** (const UnsignedIndex_t a_initial_size)
- **SmallVector** (const UnsignedIndex_t a_initial_size, const T &a_value)
- **SmallVector** (std::initializer_list< T > init)
- **T & operator[]** (const UnsignedIndex_t a_index)
- **const T & operator[]** (const UnsignedIndex_t a_index) const
- **UnsignedIndex_t size** (void) const
- **UnsignedIndex_t capacity** (void) const
- **void resize** (const UnsignedIndex_t a_count, T val=T())
- **void reserve** (const UnsignedIndex_t a_count)
- **void assign** (const UnsignedIndex_t a_count, const T &a_value)
- **T & front** (void)
- **const T & front** (void) const
- **T & back** (void)
- **const T & back** (void) const
- **bool empty** (void) const
- **bool full** (void) const
- **void pop_back** (void)
- **void insert** (const_iterator a_pos, const T &a_object)
- **void emplace** (const_iterator a_pos, T &&a_object)
- **void push_back** (const T &a_object)
- **void emplace_back** (T &&a_object)
- **void erase** (iterator a_pos)
- **void erase** (iterator a_start, iterator a_end)
- **SmallVector** (const [SmallVector](#) &a_rhs)
- **SmallVector & operator=** (const [SmallVector](#) &a_rhs)
- **iterator begin** (void) noexcept
- **const_iterator begin** (void) const noexcept
- **const_iterator cbegin** (void) const noexcept
- **iterator end** (void) noexcept
- **const_iterator end** (void) const noexcept
- **const_iterator cend** (void) const noexcept

Static Public Member Functions

- static constexpr UnsignedIndex_t **staticAllocationCapacity** (void)

Private Types

- using **BaseVector** = std::vector< T, [short_alloc::short_alloc](#)< T, static_cast< std::size_t >(BuffSizeInElements) *sizeof(T), alignof(T)> >

Private Attributes

- **BaseVector::allocator_type::arena_type memory_pool_m**
- **BaseVector vector_m**

The documentation for this class was generated from the following file:

- small_vector.h

6.872 IRL::StackVector< ObjectType, kMaxSize > Class Template Reference

Public Types

- using **iterator** = typename std::array< ObjectType, kMaxSize >::iterator
- using **const_iterator** = typename std::array< ObjectType, kMaxSize >::const_iterator

Public Member Functions

- template<UnsignedIndex_t kOtherMaxSize>
StackVector (const [StackVector](#)< ObjectType, kOtherMaxSize > &other)
- template<UnsignedIndex_t kOtherMaxSize>
StackVector & **operator=** (const [StackVector](#)< ObjectType, kOtherMaxSize > &other)
- ObjectType & **operator[]** (const UnsignedIndex_t a_index)
- const ObjectType & **operator[]** (const UnsignedIndex_t a_index) const
- UnsignedIndex_t **size** (void) const
- UnsignedIndex_t **maxSize** (void) const
- void **resize** (const UnsignedIndex_t a_count)
- void **assign** (const UnsignedIndex_t a_count, const ObjectType &a_value)
- ObjectType & **front** (void)
- const ObjectType & **front** (void) const
- ObjectType & **back** (void)
- const ObjectType & **back** (void) const
- bool **empty** (void) const
- bool **full** (void) const
- void **pop_back** (void)
- void **push_back** (const ObjectType &a_object)
- void **erase** (iterator a_pos)
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept

Private Member Functions

- void **incrementSize** (void)
- void **decrementSize** (void)

Private Attributes

- std::array< ObjectType, kMaxSize > **object_container_m**
- UnsignedIndex_t **size_m**

The documentation for this class was generated from the following file:

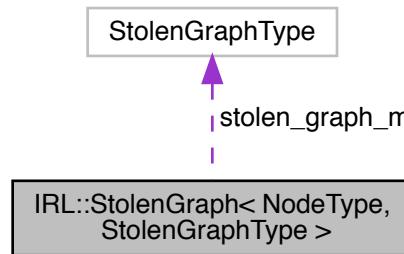
- stack_vector.h

6.873 IRL::StolenGraph< NodeType, StolenGraphType > Class Template Reference

Class to inherit from that will utilize an already set up graph and mask some of its behaviors by shadowing in the `NodeType` class. In order to have a consistent interface with other graph classes, the `NodeType` class that inherits `StolenGraph` (which works due to CRTP) must implement a `NodeType getNeighbor(const UnsignedIndex_t a_neighbor_index) const` method See [planar_reconstruction/localizer_link_from_localized_separator_link.h](#) as an example.

```
#include <stolen_graph.h>
```

Collaboration diagram for IRL::StolenGraph< NodeType, StolenGraphType >:



Public Member Functions

- `StolenGraph (const StolenGraphType *a_stolen_graph)`
Default constructor.
- `UnsignedIndex_t getNumberOfEdges (void) const`
Return the number of edges. Note, some might be to nowhere, meaning nullptr.
- `UnsignedIndex_t getId (void) const`
- `bool isIdSet (void) const`
- `bool hasNeighbor (const UnsignedIndex_t a_neighbor_index) const`
- `const stolen_node_type & getStolenNeighbor (const UnsignedIndex_t a_neighbor_index) const`
Return neighboring LocalizedSeparator.
- `const stolen_node_type * getNeighborAddress (const UnsignedIndex_t a_neighbor_index) const`
- `const stolen_node_type * getNodeMemoryAddress (void) const`
- `const stolen_node_type & getStolenGraphNode (void) const`
- `~StolenGraph (void)=default`
Default destructor.

Private Types

- using `stolen_node_type = typename StolenGraphType::node_type`

Private Attributes

- `const StolenGraphType * stolen_graph_m`

6.873.1 Detailed Description

```
template<class NodeType, class StolenGraphType>
class IRL::StolenGraph< NodeType, StolenGraphType >
```

Class to inherit from that will utilize an already set up graph and mask some of its behaviors by shadowing in the `NodeType` class. In order to have a consistent interface with other graph classes, the `NodeType` class that inherits `StolenGraph` (which works due to CRTP) must implement a `NodeType getNeighbor(const UnsignedIndex<=ta_neighbor_index) const` method See [planar_reconstruction/localizer_link_from_localized_separator_link.h](#) as an example.

The documentation for this class was generated from the following file:

- `stolen_graph.h`

6.874 IRL::StoredCappedDodecahedron< VertexType > Class Template Reference

Inheritance diagram for `IRL::StoredCappedDodecahedron< VertexType >`:



Collaboration diagram for `IRL::StoredCappedDodecahedron< VertexType >`:



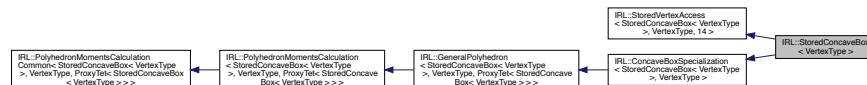
Additional Inherited Members

The documentation for this class was generated from the following file:

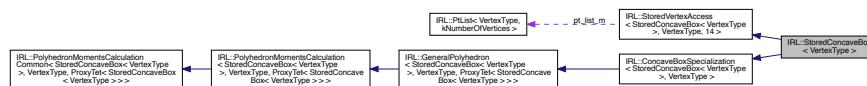
- `capped_dodecahedron.h`

6.875 IRL::StoredConcaveBox< VertexType > Class Template Reference

Inheritance diagram for IRL::StoredConcaveBox< VertexType >:



Collaboration diagram for IRL::StoredConcaveBox< VertexType >:



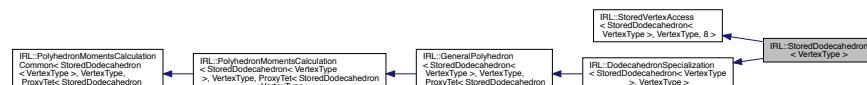
Additional Inherited Members

The documentation for this class was generated from the following file:

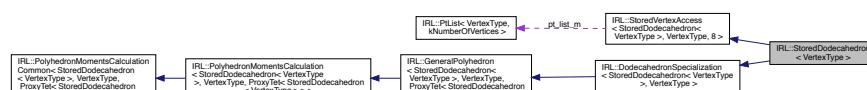
- concave_box.h

6.876 IRL::StoredDodecahedron< VertexType > Class Template Reference

Inheritance diagram for IRL::StoredDodecahedron< VertexType >:



Collaboration diagram for IRL::StoredDodecahedron< VertexType >:



Additional Inherited Members

The documentation for this class was generated from the following file:

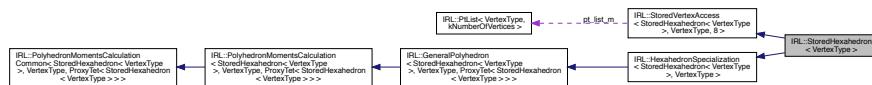
- dodecahedron.h

6.877 IRL::StoredHexahedron< VertexType > Class Template Reference

Inheritance diagram for IRL::StoredHexahedron< VertexType >:



Collaboration diagram for IRL::StoredHexahedron< VertexType >:



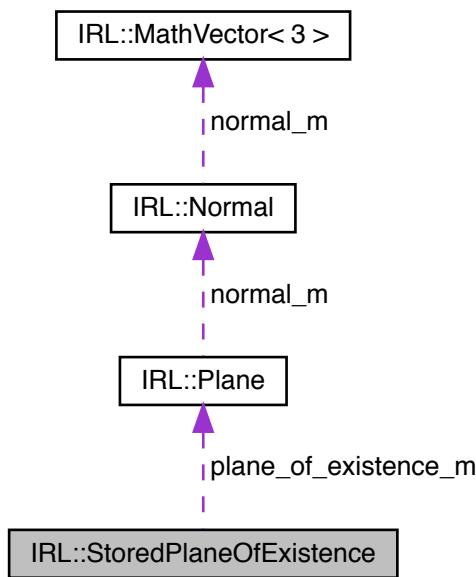
Additional Inherited Members

The documentation for this class was generated from the following file:

- hexahedron.h

6.878 IRL::StoredPlaneOfExistence Class Reference

Collaboration diagram for IRL::StoredPlaneOfExistence:



Public Member Functions

- **StoredPlaneOfExistence** (const [Plane](#) &a_plane_of_existence)
- void **setPlaneOfExistence_derived** (const [Plane](#) &a_plane)
- const [Plane](#) & **getPlaneOfExistence_derived** (void) const

Private Attributes

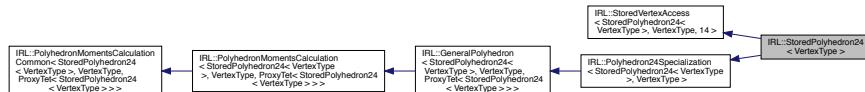
- [Plane](#) **plane_of_existence_m**

The documentation for this class was generated from the following file:

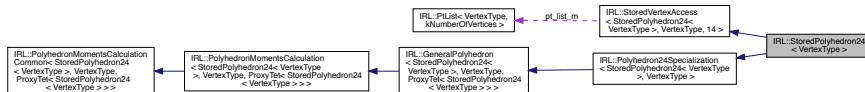
- stored_plane_of_existence.h

6.879 IRL::StoredPolyhedron24< VertexType > Class Template Reference

Inheritance diagram for IRL::StoredPolyhedron24< VertexType >:



Collaboration diagram for IRL::StoredPolyhedron24< VertexType >:



Additional Inherited Members

The documentation for this class was generated from the following file:

- polyhedron_24.h

6.880 IRL::StoredRectangularCuboid< VertexType > Class Template Reference

Inheritance diagram for IRL::StoredRectangularCuboid< VertexType >:



Collaboration diagram for IRL::StoredRectangularCuboid< VertexType >:



Static Public Member Functions

- static [StoredRectangularCuboid fromBoundingPts](#) (const [Pt &a_min_point](#), const [Pt &a_max_point](#))

Private Member Functions

- [StoredRectangularCuboid](#) (const [Pt &a_min_point](#), const [Pt &a_max_point](#))

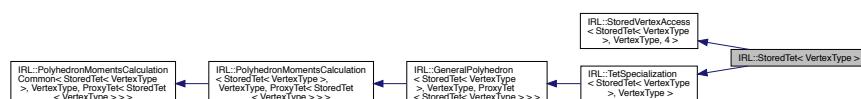
Additional Inherited Members

The documentation for this class was generated from the following file:

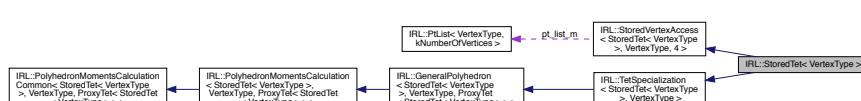
- [rectangular_cuboid.h](#)

6.881 IRL::StoredTet< VertexType > Class Template Reference

Inheritance diagram for IRL::StoredTet< VertexType >:



Collaboration diagram for IRL::StoredTet< VertexType >:



Public Member Functions

- template<class GeometryType>
StoredTet (const [ProxyTet< GeometryType >](#) &a_tet_proxy)
- template<class GeometryType>
StoredTet & **operator=** (const [ProxyTet< GeometryType >](#) &a_tet_proxy)

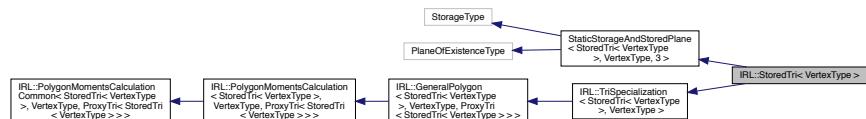
Additional Inherited Members

The documentation for this class was generated from the following file:

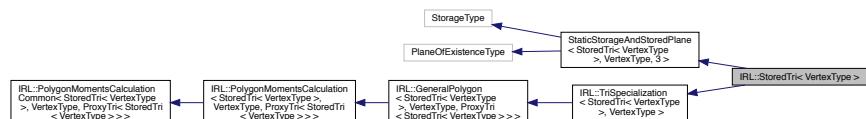
- tet.h

6.882 IRL::StoredTri< VertexType > Class Template Reference

Inheritance diagram for IRL::StoredTri< VertexType >:



Collaboration diagram for IRL::StoredTri< VertexType >:



Public Member Functions

- template<class GeometryType>
StoredTri (const [ProxyTri< GeometryType >](#) &a_tri_proxy)
- template<class GeometryType>
StoredTri & **operator=** (const [ProxyTri< GeometryType >](#) &a_tri_proxy)

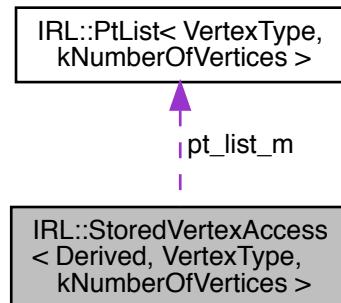
Additional Inherited Members

The documentation for this class was generated from the following file:

- tri.h

6.883 IRL::StoredVertexAccess< Derived, VertexType, kNumberOfVertices > Class Template Reference

Collaboration diagram for IRL::StoredVertexAccess< Derived, VertexType, kNumberOfVertices >:



Public Member Functions

- **StoredVertexAccess** (std::initializer_list< VertexType > a_list)
- `VertexType & access` (const UnsignedIndex_t a_index)
- `const VertexType & access` (const UnsignedIndex_t a_index) const
- `LargeOffsetIndex_t getSerializedSize` (void) const

Return size of the serialized Polyhedron.
- `void serialize` (ByteBuffer *a_buffer) const

Serialize and pack the Polyhedron.
- `void unpackSerialized` (ByteBuffer *a_buffer)

Unpack the polyhedron.

Static Public Member Functions

- static Derived `fromRawPtPointer` (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts)

Construct n-pts form array of points.
- static Derived `fromRawDoublePointer` (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs)
- static constexpr UnsignedIndex_t `getNumberOfVerticesInObject` (void)

Return the number of vertices in this polygon.

Static Public Attributes

- static constexpr UnsignedIndex_t **number_of_vertices** = kNumberOfVertices

Protected Member Functions

- **StoredVertexAccess** (const UnsignedIndex_t a_number_of_pts, const VertexType *a_array_of_pts)
Construct n-pts form array of pts.
- **StoredVertexAccess** (const UnsignedIndex_t a_number_of_pts, const double *a_array_of_locs)
Construct n-pts form array of doubles.

Private Attributes

- **PtList< VertexType, kNumberOfVertices > pt_list_m**

The documentation for this class was generated from the following file:

- stored_vertex_access.h

6.884 IRL::StrongWolfeConditionLineSearch Class Reference

Implementation of line search algorithm from "Line search algorithms with guaranteed sufficient decrease" by Jorge More and David Thuente, ACM Transactions on Mathematical Software, 1994. Some implementation details (such as the zoom function) are also taken from the book Numerical Optimization by Jorge Nocedal and Stephen Wright, 2006.

```
#include <bfsgs.h>
```

Public Member Functions

- template<class OptimizingClass , class EigenMatrix >
EigenMatrix solve (OptimizingClass *a_otype, const EigenMatrix &a_search_direction, const double a_step_size_max)

Private Member Functions

- template<class OptimizingClass , class EigenMatrix >
EigenMatrix zoom (OptimizingClass *a_otype, const EigenMatrix &a_search_direction, double a_step_size_lower, double a_step_size_upper, double a_function_value_lower, double a_function_value_upper, double a_gradient_lower, double a_gradient_upper)

Static Private Member Functions

- template<class OptimizingClass , class EigenMatrix >
static double calculateGradientOfCostFunction (OptimizingClass *a_otype, const double a_step_size, const EigenMatrix &a_search_direction, const double a_step_size_delta)
- **static double getNextTargetStep** (const double a_step_size_lower, const double a_step_size_target, const double a_step_size_upper, const double a_function_value_lower, const double a_function_value_target, const double a_function_value_upper, const double a_gradient_lower, const double a_gradient_target, const double a_gradient_upper)
- **static double cubicInterpolation** (const double a_step_size_lower, const double a_step_size_upper, const double a_function_value_lower, const double a_function_value_upper, const double a_gradient_lower, const double a_gradient_upper)
- **static double quadraticInterpolationWithTwoFunctions** (const double a_step_size_lower, const double a_step_size_upper, const double a_function_value_lower, const double a_function_value_upper, const double a_gradient_lower)
- **static double quadraticInterpolationWithTwoDerivatives** (const double a_step_size_lower, const double a_step_size_upper, const double a_function_value_lower, const double a_gradient_lower, const double a_gradient_upper)

Private Attributes

- double **initial_function_value_m**
- double **initial_function_gradient_m**

Static Private Attributes

- static constexpr UnsignedIndex_t **MAX_ITER** = 10
- static constexpr double **C1** = 1.0e-4
- static constexpr double **C2** = 0.9
- static constexpr double **GRADIENT_STEP** = 1.0e-8

6.884.1 Detailed Description

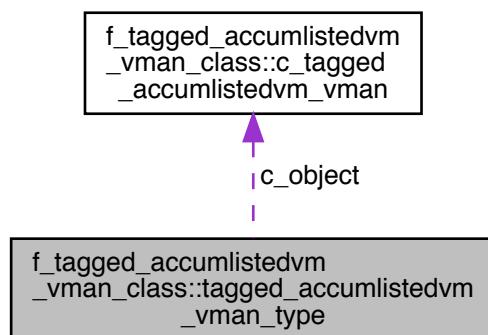
Implementation of line search algorithm from "Line search algorithms with guaranteed sufficient decrease" by Jorge More and David Thuente, ACM Transactions on Mathematical Software, 1994. Some implementation details (such as the zoom function) are also taken from the book Numerical Optimization by Jorge Nocedal and Stephen Wright, 2006.

The documentation for this class was generated from the following file:

- bfgs.h

6.885 f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_type Type Reference

Collaboration diagram for f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_type:



Public Member Functions

- final **tagged_accumlistedvm_vman_class_delete**

Private Attributes

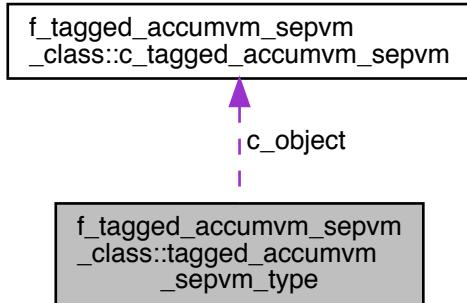
- type([c_tagged_accumlistedvm_vman](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_tagged_accumlistedvm_vman_class.f90](#)

6.886 f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_type Type Reference

Collaboration diagram for f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_type:



Public Member Functions

- final **tagged_accumvm_sepvm_class_delete**

Private Attributes

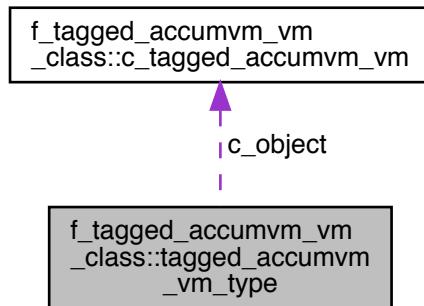
- type([c_tagged_accumvm_sepvm](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_tagged_accumvm_sepvm_class.f90](#)

6.887 f_tagged_accumvm_vm_class::tagged_accumvm_vm_type Type Reference

Collaboration diagram for f_tagged_accumvm_vm_class::tagged_accumvm_vm_type:



Public Member Functions

- final `tagged_accumvm_vm_class_delete`

Private Attributes

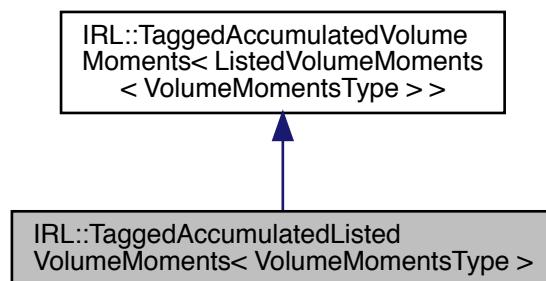
- type(`c_tagged_accumvm_vm`), private `c_object`

The documentation for this type was generated from the following file:

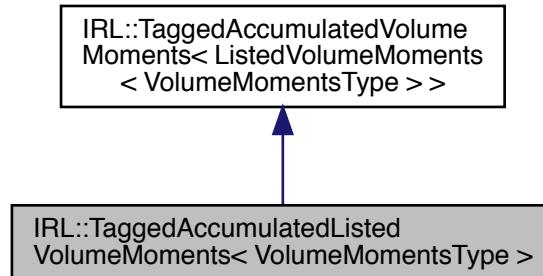
- `f_tagged_accumvm_vm_class.f90`

6.888 IRL::TaggedAccumulatedListedVolumeMoments< VolumeMomentsType > Class Template Reference

Inheritance diagram for IRL::TaggedAccumulatedListedVolumeMoments< VolumeMomentsType >:



Collaboration diagram for IRL::TaggedAccumulatedListedVolumeMoments< VolumeMomentsType >:



Public Types

- using **contained_type** = VolumeMomentsType

Public Member Functions

- [TaggedAccumulatedListedVolumeMoments](#) (void)=default
Default constructor.
- [~TaggedAccumulatedListedVolumeMoments](#) (void)=default
Default destructor.

The documentation for this class was generated from the following file:

- tagged_accumulated_listed_volume_moments.h

6.889 IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType > Class Template Reference

Classes

- struct [TaggedMoments](#)

Public Types

- using **contained_type** = VolumeMomentsType

Public Member Functions

- VolumeMomentsType & **operator[]** (const UnsignedIndex_t a_tag)
- const VolumeMomentsType & **operator[]** (const UnsignedIndex_t a_tag) const
- VolumeMomentsType & **getMomentsForIndex** (const UnsignedIndex_t a_index)
- const VolumeMomentsType & **getMomentsForIndex** (const UnsignedIndex_t a_index) const
- UnsignedIndex_t **getTagForIndex** (const UnsignedIndex_t a_index) const
- **TaggedAccumulatedVolumeMoments** & **operator+=** (const **TaggedAccumulatedVolumeMoments** &a_other)
- bool **isTagKnown** (const UnsignedIndex_t a_tag) const
- bool **isTagNew** (const UnsignedIndex_t a_tag) const
- void **normalizeByVolume** (void)

Normalize entire vector by volume.
- void **multiplyByVolume** (void)

Normalize entire vector by volume.
- **TaggedAccumulatedVolumeMoments** & **operator=** (const double a_value)

Set all moments in the list equal to a_value.
- UnsignedIndex_t **size** (void) const
- void **addNewTaggedAddress** (const UnsignedIndex_t a_tag)
- void **clear** (void)
- iterator **begin** (void) noexcept
- const_iterator **begin** (void) const noexcept
- const_iterator **cbegin** (void) const noexcept
- iterator **end** (void) noexcept
- const_iterator **end** (void) const noexcept
- const_iterator **cend** (void) const noexcept

Private Types

- using **iterator** = typename std::vector<**TaggedMoments**>::iterator
- using **const_iterator** = typename std::vector<**TaggedMoments**>::const_iterator

Private Attributes

- std::vector<**TaggedMoments**> **accumulated_moments_m**
- std::unordered_map<UnsignedIndex_t, UnsignedIndex_t> **tag_to_vector_index_m**

Static Private Attributes

- static constexpr std::size_t **initial_bucket_count** = 20

The documentation for this class was generated from the following file:

- tagged_accumulated_volume_moments.h

6.890 IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType >::TaggedMoments Struct Reference

Public Attributes

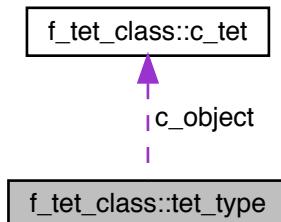
- VolumeMomentsType **volume_moments_m**
- UnsignedIndex_t **tag_m**

The documentation for this struct was generated from the following file:

- `tagged_accumulated_volume_moments.h`

6.891 f_tet_class::tet_type Type Reference

Collaboration diagram for `f_tet_class::tet_type`:



Public Member Functions

- final **tet_class_delete**

Private Attributes

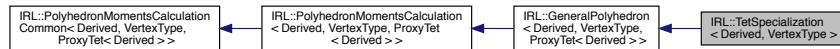
- type([c_tet](#)), private **c_object**

The documentation for this type was generated from the following file:

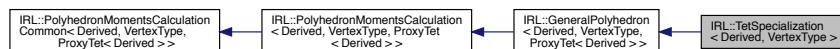
- [f_tet_class.f90](#)

6.892 IRL::TetSpecialization< Derived, VertexType > Class Template Reference

Inheritance diagram for IRL::TetSpecialization< Derived, VertexType >:



Collaboration diagram for IRL::TetSpecialization< Derived, VertexType >:



Public Member Functions

- `ProxyTet< Derived > getSimplexFromDecomposition (const UnsignedIndex_t a_tet) const`
- `HalfEdgePolyhedron< VertexType > generateHalfEdgeVersion (void) const`
- template<class HalfEdgePolyhedronType>
`void setHalfEdgeVersion (HalfEdgePolyhedronType *a_half_edge_version) const`
- `PlanarLocalizer getLocalizer (void) const`
Const version of getPlanarRepresentation.
- `Volume calculateAbsoluteVolume (void) const`
Calculate and return signed volume of the tet. See Owkes & Desjardins, JCP, 2014.
- `Volume calculateVolume (void) const`
Calculate and return volume of the tet.
- `double calculateSign (void) const`
Calculate sign for the tet See Owkes & Desjardins, JCP, 2014.
- `Pt calculateCentroid (void) const`
Calculate and return centroid of the tet.
- `VolumeMoments calculateMoments () const`
Calculate and return volume weighted VolumeMoments.

Static Public Member Functions

- `static constexpr UnsignedIndex_t getNumberOfSimplicesInDecomposition (void)`
Returns the tet.
- `static constexpr std::array< UnsignedIndex_t, 4 > getSimplexIndicesFromDecomposition (const UnsignedIndex_t a_tet)`

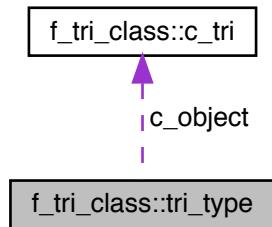
Additional Inherited Members

The documentation for this class was generated from the following file:

- `tet.h`

6.893 f_tri_class::tri_type Type Reference

Collaboration diagram for f_tri_class::tri_type:



Public Member Functions

- final **tri_class_delete**

Private Attributes

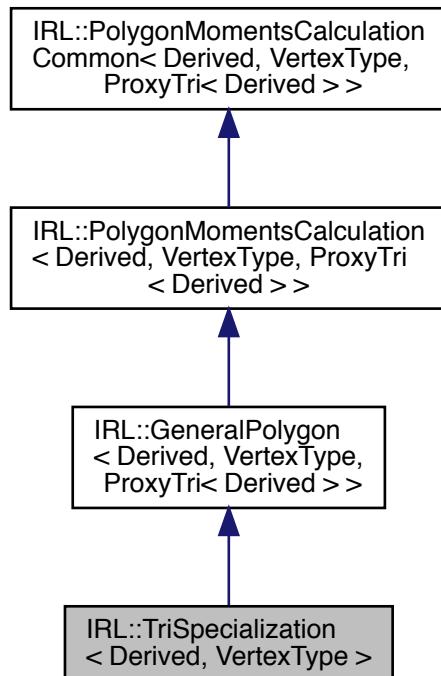
- type([c_tri](#)), private **c_object**

The documentation for this type was generated from the following file:

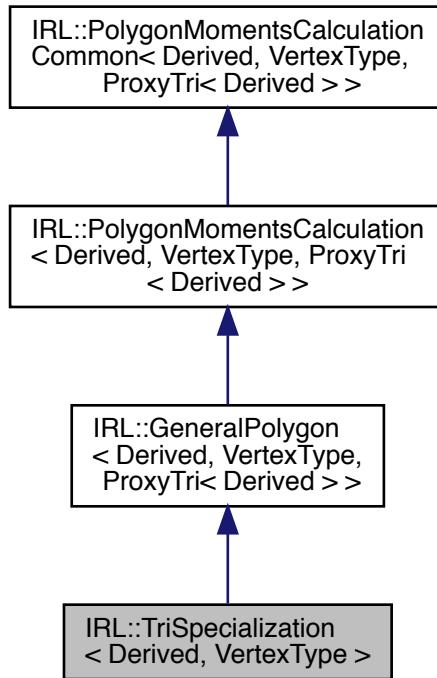
- [f_tri_class.f90](#)

6.894 IRL::TriSpecialization< Derived, VertexType > Class Template Reference

Inheritance diagram for IRL::TriSpecialization< Derived, VertexType >:



Collaboration diagram for IRL::TriSpecialization< Derived, VertexType >:



Public Types

- using **pt_type** = VertexType

Public Member Functions

- **ProxyTri< Derived >** [getSimplexFromDecomposition](#) (const UnsignedIndex_t a_tri_number_to_get) const
Const version of getSimplexFromDecomposition.
- **HalfEdgePolygon< pt_type >** [generateHalfEdgeVersion](#) (void) const
- template<class HalfEdgePolygonType >
void [setHalfEdgeVersion](#) (HalfEdgePolygonType *a_half_edge_version) const
- void [reversePtOrdering](#) (void)
- **Pt** [calculateCentroid](#) (void) const
- **Volume** [calculateVolume](#) (void) const
- **Volume** [calculateAbsoluteVolume](#) (void) const
- **VolumeMomentsAndNormal** [calculateVolumeMomentsAndNormal](#) (void) const
- **VolumeMoments** [calculateMoments](#) (void) const
- double [calculateSign](#) (void) const

Static Public Member Functions

- static constexpr UnsignedIndex_t [getNumberOfSimplicesInDecomposition](#) (void)
Returns the triangle.
- static constexpr std::array< UnsignedIndex_t, 3 > [getSimplexIndicesFromDecomposition](#) (UnsignedIndex_t a_tri_number_to_get)

The documentation for this class was generated from the following file:

- tri.h

6.895 IRL::UnDirectedGraphNode< NodeType > Class Template Reference

Class to inherit from to provide linking between planes in a planar reconstruction to other reconstructions, where the linking exists PER PLANE.

```
#include <un_directed_graph_node.h>
```

Public Types

- using **node_type** = NodeType

Public Member Functions

- [UnDirectedGraphNode](#) (void)
Default constructor.
- void [setEdgeConnectivity](#) (const UnsignedIndex_t a_edge_index, const NodeType *a_neighbor_ptr)
Creates an edge for the supplied a_edge_index. If this is greater than the current number of edges, the edge storage will be expanded with new elements (not at a_edge_index) given nullptr, indicating a edge to nowhere.
- UnsignedIndex_t [getNumberOfEdges](#) (void) const
Return the number of edges. Note, some might be to nowhere, meaning nullptr.
- void [setId](#) (const UnsignedIndex_t a_unique_id)
- UnsignedIndex_t [getId](#) (void) const
- bool [isIdSet](#) (void) const
- bool [hasNeighbor](#) (const UnsignedIndex_t a_neighbor_index) const
- const NodeType & [getNeighbor](#) (const UnsignedIndex_t a_neighbor_index) const
Return neighboring LocalizedSeparator.
- const NodeType * [getNeighborAddress](#) (const UnsignedIndex_t a_neighbor_index) const
- const NodeType * [getNodeMemoryAddress](#) (void) const
- [~UnDirectedGraphNode](#) (void)=default
Default destructor.

Private Attributes

- std::vector< const NodeType * > **edge_list_m**
- UnsignedIndex_t **id_m**

6.895.1 Detailed Description

```
template<class NodeType>
class IRL::UnDirectedGraphNode< NodeType >
```

Class to inherit from to provide linking between planes in a planar reconstruction to other reconstructions, where the linking exists PER PLANE.

The documentation for this class was generated from the following file:

- un_directed_graph_node.h

6.896 IRL::UnitQuaternion Class Reference

Unit quaternion to be used to perform rotations.

```
#include <unit_quaternion.h>
```

Public Member Functions

- **UnitQuaternion** (const double a_rotation_amount_in_radians, const **Normal** &a_rotation_axis)
Construct quaternion given rotation amount (in radians) and rotation axis.
- const double & **operator[]** (const UnsignedIndex_t a_elem) const
Const access through overload[]
- void **normalize** (void)
Normalize the unit quaternion.
- double **magnitude** (void) const
Return magnitude of the quaternion.
- **UnitQuaternion inverse** (void) const
Return a copy of the inverse UnitQuaternion.
- **UnitQuaternion operator*** (const **UnitQuaternion** &a_unit_quaternion) const
Compile unit quaternions to perform successive rotations.
- **Normal operator*** (const **Normal** &a_normal) const
Rotate a normal.
- **ReferenceFrame operator*** (const **ReferenceFrame** &a_reference_frame) const
Rotate a reference frame.
- **~UnitQuaternion** (void)=default
Default destructor.

Static Public Member Functions

- static **UnitQuaternion fromFourElements** (double a_q0, double a_q1, double a_q2, double a_q3)
- static **UnitQuaternion fromFourElementsNormalized** (double a_q0, double a_q1, double a_q2, double a_q3)

Private Member Functions

- double & [operator\[\]](#) (const UnsignedIndex_t a_elem)
Access through overload[]
- [UnitQuaternion](#) (double a_q0, double a_q1, double a_q2, double a_q3)
Constructor given 4 doubles.

Private Attributes

- std::array< double, 4 > [quat_m](#)
Storage for the 4 elements of a quaternion.

6.896.1 Detailed Description

Unit quaternion to be used to perform rotations.

The documentation for this class was generated from the following file:

- unit_quaternion.h

6.897 f_serializer::unpackandstore Interface Reference

Public Member Functions

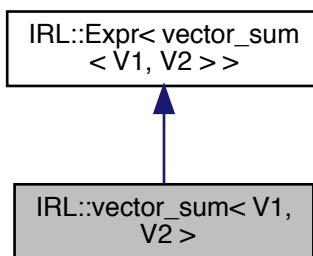
- subroutine [unpackandstore_planarseparator_bytbuffer](#) (a_separator, a_byte_buffer)

The documentation for this interface was generated from the following file:

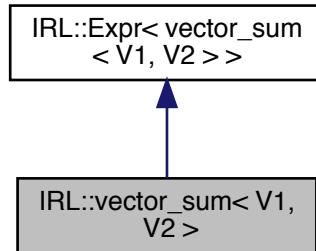
- [f_serializer.f90](#)

6.898 IRL::vector_sum< V1, V2 > Class Template Reference

Inheritance diagram for IRL::vector_sum< V1, V2 >:



Collaboration diagram for IRL::vector_sum< V1, V2 >:



Public Types

- using **value_type** = typename std::common_type< typename V1::value_type, typename V2::value_type >::type

Public Member Functions

- **vector_sum** (const V1 &a_v1, const V2 &a_v2)
- value_type **operator[]** (const UnsignedIndex_t i) const

Private Types

- using **self** = [vector_sum](#)

Private Member Functions

- void **check_index** (const UnsignedIndex_t i) const

Private Attributes

- const V1 & **v1_m**
- const V2 & **v2_m**

Friends

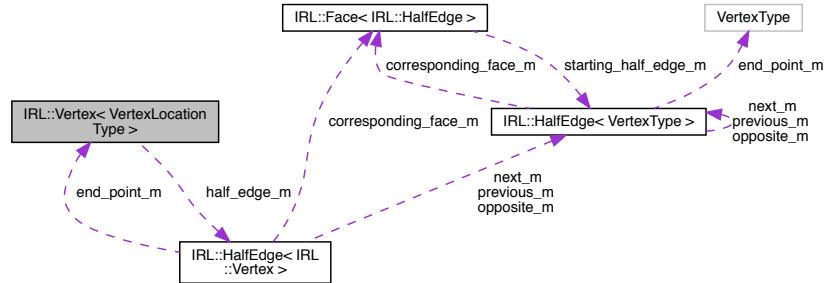
- UnsignedIndex_t **size** (const [self](#) &x)

The documentation for this class was generated from the following file:

- [helper.h](#)

6.899 IRL::Vertex< VertexLocationType > Class Template Reference

Collaboration diagram for IRL::Vertex< VertexLocationType >:



Public Types

- using `vertex_location_type = VertexLocationType`

Public Member Functions

- `Vertex (VertexLocationType *a_location)`
- `void setHalfEdge (HalfEdge< Vertex > *a_half_edge)`
- `HalfEdge< Vertex > * getHalfEdge (void)`
- `const HalfEdge< Vertex > * getHalfEdge (void) const`
- `void setVertexLocation (VertexLocationType *a_location)`
- `VertexLocationType * getVertexLocation (void)`
- `const VertexLocationType::pt_type & getLocation (void) const`
- `void calculateDistanceToPlane (const Plane &a_plane)`
- `double getDistance (void) const`
- `void markToBeClipped (void)`
- `void markToBeNotClipped (void)`
- `bool isClipped (void) const`
- `bool isNotClipped (void) const`
- `void setAsUnnecessaryToSeek (void)`
- `void setToSeek (void)`
- `bool needsToSeek (void) const`
- `bool doesNotNeedToSeek (void) const`
- `bool checkValidHalfEdgeCycle (void) const`

Private Attributes

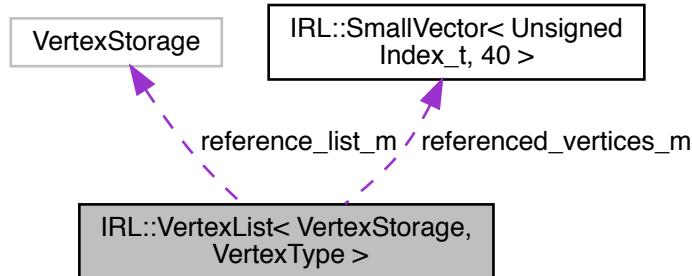
- `VertexLocationType * vertex_location_m`
- `HalfEdge< Vertex > * half_edge_m`
- `bool is_clipped_m`

The documentation for this class was generated from the following file:

- `half_edge.h`

6.900 IRL::VertexList< VertexStorage, VertexType > Class Template Reference

Collaboration diagram for IRL::VertexList< VertexStorage, VertexType >:



Public Types

- using **iterator** = `IteratorThroughBracketOperator< VertexList< VertexStorage, VertexType > >`
- using **const_iterator** = `ConstIteratorThroughBracketOperator< VertexList< VertexStorage, VertexType > >`
- using **value_t** = `VertexType`

Public Member Functions

- `VertexList (VertexStorage &a_reference)`
- `void assumeWholeOfReference (void)`
- `VertexList (const VertexList &a_other)`
- `VertexList (VertexList &&a_other)`
- `VertexList & operator= (const VertexList &a_other)`
- `VertexList & operator= (VertexList &&a_other)`
- `VertexType & operator[] (const UnsignedIndex_t a_index)`
- `const VertexType & operator[] (const UnsignedIndex_t a_index) const`
- `double getDistance (const UnsignedIndex_t a_index) const`
- `void setDistance (const UnsignedIndex_t a_index, const double a_distance) const`
- `UnsignedIndex_t & getVertexIndex (const UnsignedIndex_t a_index)`
- `const UnsignedIndex_t & getVertexIndex (const UnsignedIndex_t a_index) const`
- `void rePointVertex (const UnsignedIndex_t a_index, const UnsignedIndex_t a_new_vertex_index)`
- `void push_back (const UnsignedIndex_t a_new_vertex_index)`
- `UnsignedIndex_t size (void) const`
- `void resize (const UnsignedIndex_t a_size)`
- `VertexStorage & getUnderlyingVertexStorage (void)`
- `const VertexStorage & getUnderlyingVertexStorage (void) const`
- `iterator begin (void) noexcept`
- `const_iterator begin (void) const noexcept`
- `const_iterator cbegin (void) const noexcept`
- `iterator end (void) noexcept`
- `const_iterator end (void) const noexcept`
- `const_iterator cend (void) const noexcept`

Private Attributes

- VertexStorage & **reference_list_m**
- **SmallVector**< UnsignedIndex_t, 40 > **referenced_vertices_m**

The documentation for this class was generated from the following file:

- segmented_decomposed_polytope.h

6.901 IRL::VertexLocation< PtType > Class Template Reference

Public Types

- using **pt_type** = PtType

Public Member Functions

- **VertexLocation** (const PtType &a_pt)
- void **setLocation** (const PtType &a_pt)
- const PtType & **getLocation** (void) const
- void **calculateDistanceToPlane** (const **Plane** &a_plane)
- double **getDistance** (void) const
- void **setAsUnnecessaryToSeek** (void)
- void **setToSeek** (void)
- bool **needsToSeek** (void) const
- bool **doesNotNeedToSeek** (void) const

Private Attributes

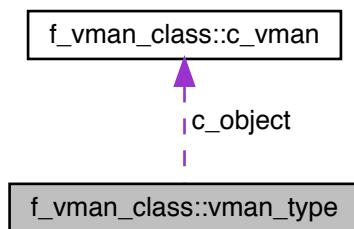
- PtType **location_m**
- double **distance_to_plane_m**
- bool **needs_to_seek_m**

The documentation for this class was generated from the following file:

- half_edge.h

6.902 f_vman_class::vman_type Type Reference

Collaboration diagram for f_vman_class::vman_type:



Public Member Functions

- final **vman_class_delete**

Private Attributes

- type([c_vman](#)), private **c_object**

The documentation for this type was generated from the following file:

- [f_vman_class.f90](#)

6.903 IRL::Volume Class Reference

A volume class which is just a double with special properties. Allows more general writing of functions.

```
#include <volume.h>
```

Public Member Functions

- [Volume \(void\)](#)
Default constructor.
- [constexpr Volume \(const double a_value\)](#)
Constructor that initializes volume to value, want to allow implicit casting of double to Volume here.
- [void multiplyByVolume \(void\)](#)
Dummy function to allow general use along with VolumeMoments.
- [void normalizeByVolume \(void\)](#)
Dummy function to allow general use along with VolumeMoments.
- [Volume & operator+= \(const Volume &a_rhs\)](#)
Overload += operator to update volume.
- [Volume & operator*= \(const double a_rhs\)](#)
*Overload *= operator to multiply by constant double.*
- [Volume & operator/= \(const double a_rhs\)](#)
Overload /= operator to divide by constant double.
- [operator double \(\) const](#)
Allow implicit conversion to double.
- [Volume & operator= \(const double a_value\)](#)
Overload assignment to assign constant value to moments.
- [~Volume \(void\)=default](#)
Default destructor.

Static Public Member Functions

- static [Volume fromScalarConstant \(const double a_value\)](#)
- template<class GeometryType >
 static [Volume calculateMoments \(GeometryType *a_geometry\)](#)
Obtain Volume from the supplied geometry.

Private Attributes

- double `volume_m`
`Volume` of something.

6.903.1 Detailed Description

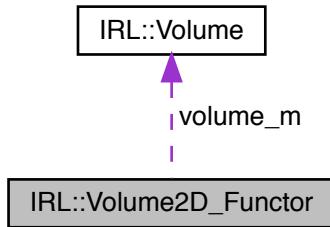
A volume class which is just a double with special properties. Allows more general writing of functions.

The documentation for this class was generated from the following file:

- `volume.h`

6.904 IRL::Volume2D_Functor Class Reference

Collaboration diagram for IRL::Volume2D_Functor:



Public Types

- using `ReturnType` = `Volume`

Public Member Functions

- template<class SimplexType >
`void operator()` (const SimplexType &a_simplex)
- `ReturnType getMoments` (void) const

Private Attributes

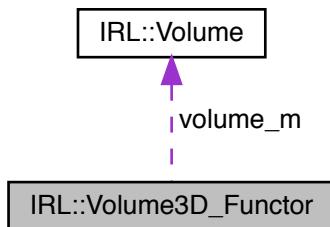
- `Volume volume_m`

The documentation for this class was generated from the following file:

- `moment_calculation_through_simplices.h`

6.905 IRL::Volume3D_Functor Class Reference

Collaboration diagram for IRL::Volume3D_Functor:



Public Types

- using **ReturnType** = [Volume](#)

Public Member Functions

- template<class SimplexType >
void **operator()** (const SimplexType &a_simplex)
- [ReturnType](#) **getMoments** (void) const

Private Attributes

- [Volume](#) **volume_m**

The documentation for this class was generated from the following file:

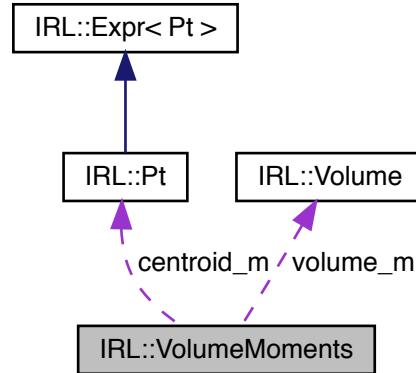
- moment_calculation_through_simplices.h

6.906 IRL::VolumeMoments Class Reference

Zeroeth (volume) and first order (centroid) geometric moments.

```
#include <volume_moments.h>
```

Collaboration diagram for IRL::VolumeMoments:



Public Member Functions

- `VolumeMoments (void)`
Default constructor.
- `constexpr VolumeMoments (const double a_volume, const Pt &a_centroid)`
Constructor that initializes volume and centroid.
- `Volume & volume (void)`
Return value of stored volume.
- `const Volume & volume (void) const`
Return const reference to stored volume.
- `Pt & centroid (void)`
Return copy of stored centroid.
- `const Pt & centroid (void) const`
Return const reference to stored centroid.
- `void normalizeByVolume (void)`
Divide the centroid by the volume.
- `void multiplyByVolume (void)`
Multiply the centroid by the volume.
- `VolumeMoments & operator+= (const VolumeMoments &a_rhs)`
Overload += operator to update moments.
- `VolumeMoments & operator*=(const double a_rhs)`
*Overload *= operator to multiply by constant double.*
- `VolumeMoments & operator/=(const double a_rhs)`
*Overload *= operator to multiply by constant double.*
- `VolumeMoments & operator= (const double a_value)`
Overload assignment to assign constant value to moments.
- `~VolumeMoments (void)=default`
Default destructor.

Static Public Member Functions

- static constexpr `VolumeMoments fromRawDoublePointer` (const double *a_list)
- static constexpr `VolumeMoments fromScalarConstant` (const double a_value)
- template<class GeometryType >
static `VolumeMoments calculateMoments` (GeometryType *a_geometry)

Obtain un-normalized VolumeMoments from the supplied geometry.

Private Member Functions

- constexpr `VolumeMoments` (const double *a_list)
Construct VolumeMoments from a list of doubles.
- constexpr `VolumeMoments` (const double a_value)
Construct that initializes volume/centroid as a value.

Private Attributes

- `Volume volume_m`
Zeroeth moment (volume).
- `Pt centroid_m`
First moment (centroid).

6.906.1 Detailed Description

Zeroeth (volume) and first order (centroid) geometric moments.

6.906.2 Constructor & Destructor Documentation

6.906.2.1 VolumeMoments()

```
constexpr IRL::VolumeMoments::VolumeMoments (
    const double * a_list ) [explicit], [private]
```

Construct `VolumeMoments` from a list of doubles.

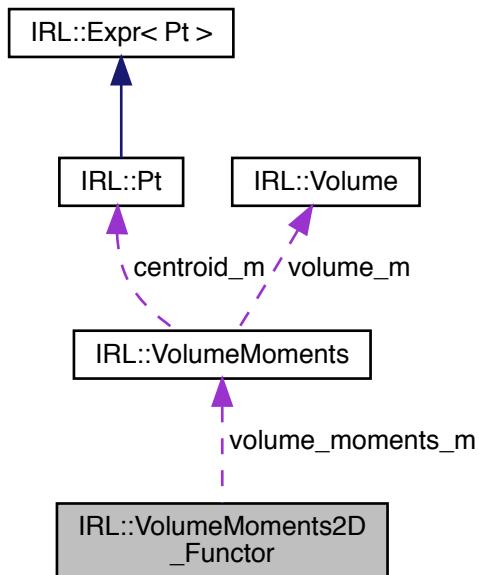
Construct `VolumeMoments` from a list of 4 doubles. The necessary order is volume, centroid_x, centroid_y, centroid_z.

The documentation for this class was generated from the following file:

- `volume_moments.h`

6.907 IRL::VolumeMoments2D_Functor Class Reference

Collaboration diagram for IRL::VolumeMoments2D_Functor:



Public Types

- using **ReturnType** = [VolumeMoments](#)

Public Member Functions

- template<class SimplexType >
void **operator()** (const SimplexType &a_simplex)
- **ReturnType** **getMoments** (void) const

Private Attributes

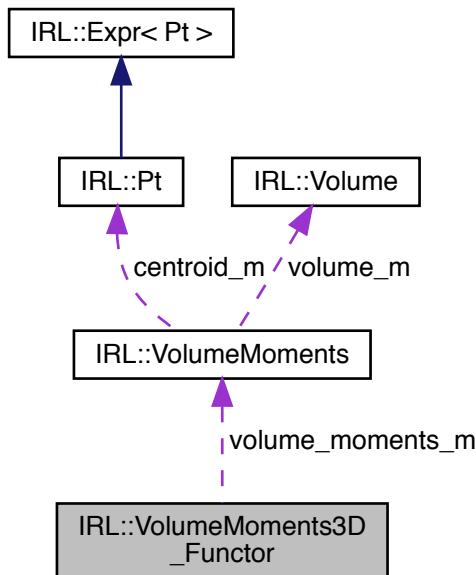
- [VolumeMoments volume_moments_m](#)

The documentation for this class was generated from the following file:

- moment_calculation_through_simplices.h

6.908 IRL::VolumeMoments3D_Functor Class Reference

Collaboration diagram for IRL::VolumeMoments3D_Functor:



Public Types

- using **ReturnType** = [VolumeMoments](#)

Public Member Functions

- template<class SimplexType >
void **operator()** (const SimplexType &a_simplex)
- [ReturnType](#) **getMoments** (void) const

Private Attributes

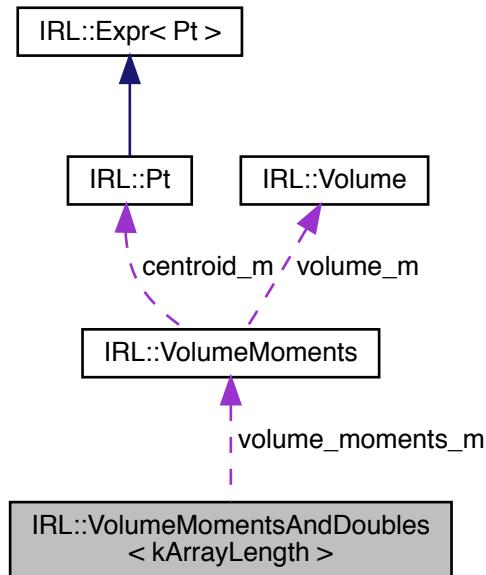
- [VolumeMoments](#) **volume_moments_m**

The documentation for this class was generated from the following file:

- moment_calculation_through_simplices.h

6.909 IRL::VolumeMomentsAndDoubles< kArrayLength > Class Template Reference

Collaboration diagram for IRL::VolumeMomentsAndDoubles< kArrayLength >:



Public Member Functions

- **VolumeMomentsAndDoubles (void)**
Default constructor.
- **VolumeMomentsAndDoubles (const double a_volume, const Pt &a_centroid, const ArrayType &a_initial_data)**
Constructor that initializes volume and centroid.
- **Volume & volume (void)**
Return value of stored volume.
- **const Volume & volume (void) const**
Return const reference to stored volume.
- **Pt & centroid (void)**
Return copy of stored centroid.
- **const Pt & centroid (void) const**
Return const reference to stored centroid.
- **ArrayType & data (void)**
Return reference to stored double array.
- **const ArrayType & data (void) const**
Return const reference to stored double array.
- **void normalizeByVolume (void)**
Divide the centroid by the volume.
- **void multiplyByVolume (void)**

- `VolumeMomentsAndDoubles & operator+= (const VolumeMomentsAndDoubles &a_rhs)`
Overload += operator to update moments.
- `VolumeMomentsAndDoubles & operator*=(const double a_rhs)`
*Overload *= operator to multiply by constant double.*
- `VolumeMomentsAndDoubles & operator/=(const double a_rhs)`
Overload /= operator to multiply by constant double.
- `VolumeMomentsAndDoubles & operator=(const double a_value)`
Overload assignment to assign constant value to moments.
- `~VolumeMomentsAndDoubles (void)=default`
Default destructor.

Static Public Member Functions

- static `VolumeMomentsAndDoubles fromRawDoublePointer (const double *a_list, const double *a_data_list)`
- static `VolumeMomentsAndDoubles fromScalarConstant (const double a_value)`
- static `VolumeMomentsAndDoubles fromScalarConstant (const double a_value, const double a_value_for_data)`
- template<class GeometryType >
 static `VolumeMomentsAndDoubles calculateMoments (GeometryType *a_geometry)`
Obtain un-normalized VolumeMoments from the supplied geometry.

Static Public Attributes

- static constexpr UnsignedIndex_t **data_length** = kArrayLength

Private Types

- using `ArrayType = std::array< double, kArrayLength >`

Private Member Functions

- `VolumeMomentsAndDoubles (const double *a_list, const double *a_data_list)`
Construct VolumeMoments from a list of doubles.
- `VolumeMomentsAndDoubles (const double a_value, const double a_value_for_data)`
Construct that initializes volume/centroid as a value.

Private Attributes

- `VolumeMoments volume_moments_m`
- `std::array< double, kArrayLength > data_moments_m`

6.909.1 Constructor & Destructor Documentation

6.909.1.1 VolumeMomentsAndDoubles()

```
template<UnsignedIndex_t kArrayLength>
IRL::VolumeMomentsAndDoubles<kArrayLength>::VolumeMomentsAndDoubles (
    const double * a_list,
    const double * a_data_list ) [explicit], [private]
```

Construct [VolumeMoments](#) from a list of doubles.

Construct [VolumeMoments](#) from a list of 4 doubles. The necessary order is volume, centroid_x, centroid_y, centroid_z.

The documentation for this class was generated from the following file:

- `volume_moments_and_doubles.h`

6.910 IRL::VolumeMomentsAndDoubles3D_Functor<kArrayLength> Class Template Reference

Public Types

- using **ReturnType** = [VolumeMomentsAndDoubles](#)<kArrayLength>

Public Member Functions

- template<class SimplexType>
void **operator()** (const SimplexType &a_simplex)
- **ReturnType** **getMoments** (void) const

Private Attributes

- **ReturnType** **volume_moments_and_doubles_m**

The documentation for this class was generated from the following file:

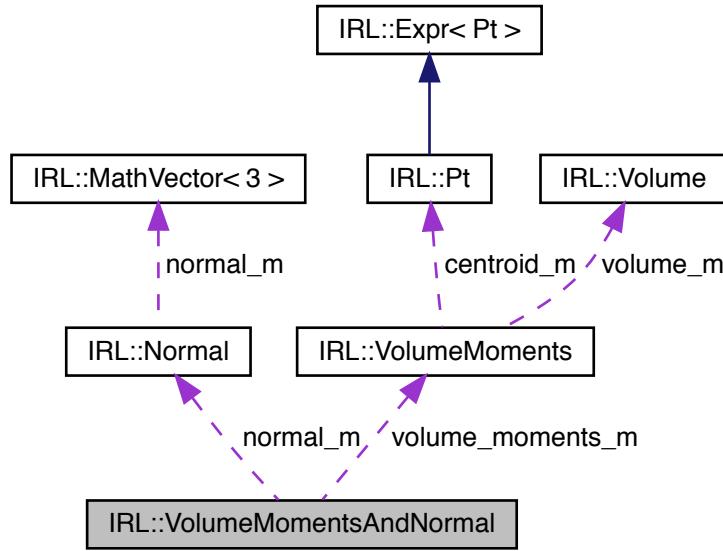
- `moment_calculation_through_simplices.h`

6.911 IRL::VolumeMomentsAndNormal Class Reference

Class that contains [VolumeMoments](#) for a triangle and the triangle's normal.

```
#include <volume_moments_and_normal.h>
```

Collaboration diagram for IRL::VolumeMomentsAndNormal:



Public Member Functions

- [VolumeMomentsAndNormal \(void\)=default](#)
Default constructor.
- [VolumeMomentsAndNormal \(const VolumeMoments &a_volume_moments, const Normal &a_normal\)](#)
Constructor given VolumeMoments and a normal.
- [VolumeMoments & volumeMoments \(void\)](#)
Return address of volume_moments_m.
- [const VolumeMoments & volumeMoments \(void\) const](#)
Return const address of volume_moments_m.
- [Normal & normal \(void\)](#)
Return normal.
- [const Normal & normal \(void\) const](#)
Return const normal.
- [void normalizeByVolume \(void\)](#)
Normalize centroid by corresponding volume.
- [void multiplyByVolume \(void\)](#)
Multiply centroid by corresponding volume.
- [VolumeMomentsAndNormal & operator+= \(const VolumeMomentsAndNormal &a_rhs\)](#)
Overload += operator to update moments.

- `VolumeMomentsAndNormal & operator*=(const double &a_rhs)`
*Overload *= operator to update moments.*
- `VolumeMomentsAndNormal & operator/=(const double &a_rhs)`
Overload /= operator to update moments.
- `VolumeMomentsAndNormal & operator=(const double a_value)`
Overload assignment to assign constant value to moments.
- `~VolumeMomentsAndNormal (void)=default`
Default destructor.

Static Public Member Functions

- template<class GeometryType >
`static VolumeMomentsAndNormal calculateMoments (GeometryType *a_geometry)`
Obtain un-normalized VolumeMoments from the supplied geometry.
- `static VolumeMomentsAndNormal fromScalarConstant (const double a_constant)`

Private Member Functions

- `VolumeMomentsAndNormal (const double a_constant)`

Private Attributes

- `VolumeMoments volume_moments_m`
- `Normal normal_m`

6.911.1 Detailed Description

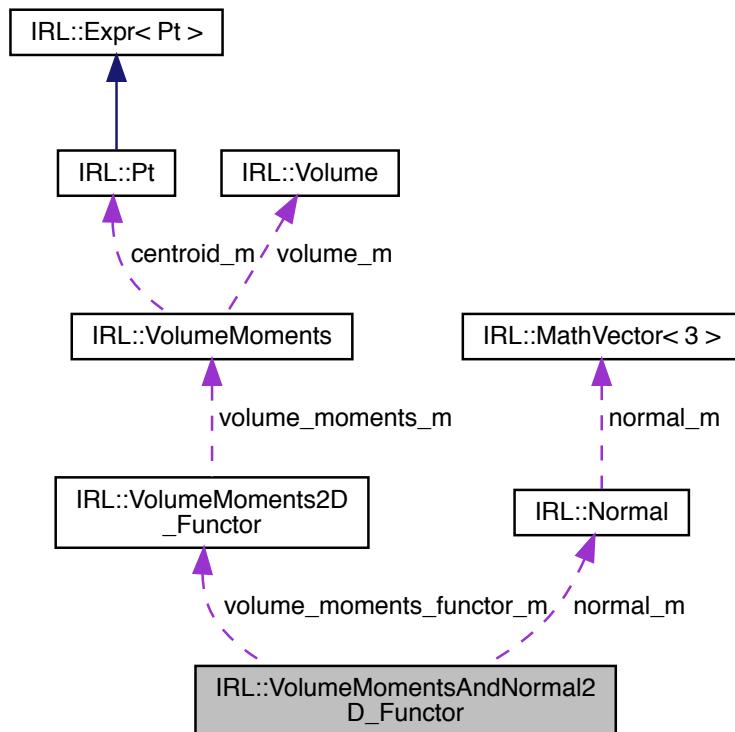
Class that contains `VolumeMoments` for a triangle and the triangle's normal.

The documentation for this class was generated from the following file:

- `volume_moments_and_normal.h`

6.912 IRL::VolumeMomentsAndNormal2D_Functor Class Reference

Collaboration diagram for IRL::VolumeMomentsAndNormal2D_Functor:



Public Types

- using **ReturnType** = [VolumeMomentsAndNormal](#)

Public Member Functions

- template<class SimplexType>
void **operator()** (const SimplexType &a_simplex)
- ReturnType** **getMoments** (void) const

Private Attributes

- VolumeMoments2D_Functor** **volume_moments_functor_m**
- Normal** **normal_m**

The documentation for this class was generated from the following file:

- moment_calculation_through_simplices.h

6.913 f_listedvm_vman_class::zeronormalcomponent Interface Reference

Public Member Functions

- subroutine **listedvm_vman_class_zeronormalcomponent** (this, a_index)

The documentation for this interface was generated from the following file:

- [f_listedvm_vman_class.f90](#)

6.914 f_dividedpolygon_class::zeropolygon Interface Reference

Public Member Functions

- subroutine **dividedpolygon_class_zeropolygon** (this)

The documentation for this interface was generated from the following file:

- [f_dividedpolygon_class.f90](#)

6.915 f_polygon_class::zeropolygon Interface Reference

Public Member Functions

- subroutine **polygon_class_zeropolygon** (this)

The documentation for this interface was generated from the following file:

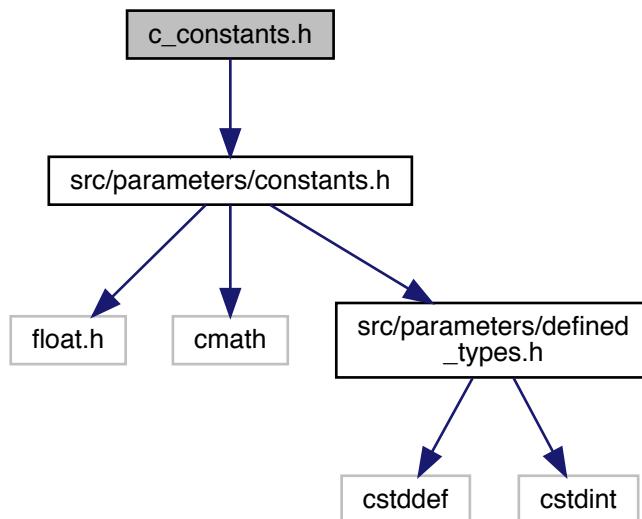
- [f_polygon_class.f90](#)

Chapter 7

File Documentation

7.1 c_constants.h File Reference

```
#include "src/parameters/constants.h"  
Include dependency graph for c_constants.h:
```



Functions

- void `c_Constants_setVolumeFractionBounds` (const double *`a_VF_low`)
Set VF_LOW and VF_HIGH defined in src/constants.cpp.
- void `c_Constants_setVolumeFractionToleranceForIterativeDistanceFinding` (const double *`a_tolerance`)
Set the volume fraction tolerance for iterative distance finding.
- void `c_Constants_setMinimumVolumeToTrack` (const double *`a_minimum_volume_to_track`)
Function to set MINIMUM_VOLUME_TO_TRACK defined in src/constants.cpp.
- void `c_Constants_setMinimumSurfaceAreaToTrack` (const double *`a_minimum_surface_area_to_track`)
Function to set MINIMUM_SURFACE_AREA_TO_TRACK defined in src/constants.cpp.

7.1.1 Detailed Description

These C-style functions are mapped to functions available in src/constants.h.

This file deals with functions that set global functions involved in the [IRL](#) library.

Individual documentation for each function is given alongside the function.

7.1.2 Function Documentation

7.1.2.1 c_Constants_setMinimumSurfaceAreaToTrack()

```
void c_Constants_setMinimumSurfaceAreaToTrack (
    const double * a_minimum_surface_area_to_track )
```

Function to set MINIMUM_SURFACE_AREA_TO_TRACK defined in src/constants.cpp.

This function sets MINIMUM_SURFACE_AREA_TO_TRACK to the value a_minimum_surface_area_to_track. MINIMUM_SURFACE_AREA_TO_TRACK is primarily used as the terminating condition for the numerical integration and subdivision of polygons, where sub-areas less than MINIMUM_SURFACE_AREA_TO_TRACK will be ignored.

Parameters

in	a_minimum_surface_area_to_track	Value to set MINIMUM_SURFACE_AREA_TO_TRACK to.
----	---------------------------------	--

7.1.2.2 c_Constants_setMinimumVolumeToTrack()

```
void c_Constants_setMinimumVolumeToTrack (
    const double * a_minimum_volume_to_track )
```

Function to set MINIMUM_VOLUME_TO_TRACK defined in src/constants.cpp.

This function sets MINIMUM_VOLUME_TO_TRACK to the value a_minimum_volume_to_track. MINIMUM_VOLUME_TO_TRACK is primarily used as the terminating condition for the numerical integration and subdivision of polyhedra, where sub-volumes less than MINIMUM_VOLUME_TO_TRACK will be ignored.

Parameters

in	a_minimum_volume_to_track	Value to set MINIMUM_VOLUME_TO_TRACK to.
----	---------------------------	--

7.1.2.3 c_Constants_setVolumeFractionBounds()

```
void c_Constants_setVolumeFractionBounds (
    const double * a_VF_low )
```

Set VF_LOW and VF_HIGH defined in src/constants.cpp.

This function sets bounds on Volume Fraction (VF), setting VF_LOW and VF_HIGH for all future computations. These are used as threshold values to terminate some computations, especially during calculations of interface reconstructions. VF_HIGH will automatically be set as 1.0 - a_VF_low in order to preserve symmetry.

Parameters

in	a_VF_low	Value to set VF_LOW.
----	----------	----------------------

7.1.2.4 c_Constants_setVolumeFractionToleranceForIterativeDistanceFinding()

```
void c_Constants_setVolumeFractionToleranceForIterativeDistanceFinding (
    const double * a_tolerance )
```

Set the volume fraction tolerance for iterative distance finding.

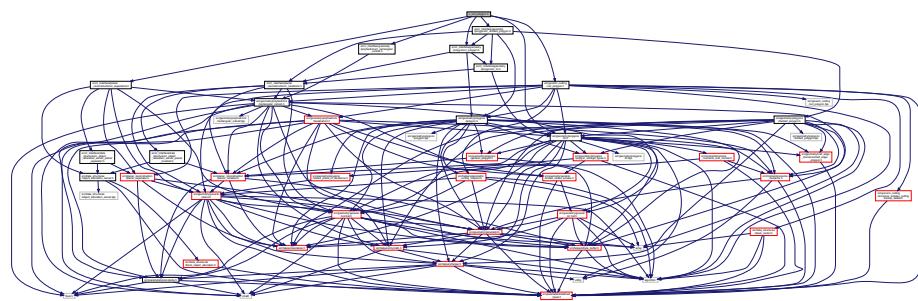
This function sets the default volume fraction tolerance to be used when an iterative distance finding routine is used. It will always be the minimum of a_tolerance and VF_LOW.

Parameters

in	a_tolerance	Default volume fraction tolerance to use during iterative distance finding.
----	-------------	---

7.2 c_cut_polygon.h File Reference

```
#include "src/c_interface/geometry/polygons/c_divided_polygon.h"
#include "src/c_interface/geometry/polygons/c_polygon.h"
#include "src/c_interface/geometry/polyhedrons/c_rectangular_cuboid.h"
#include "src/c_interface/planar_reconstruction/c_separators.h"
#include "src/generic_cutting/cut_polygon.h"
#include "src/geometry/polygons/tri.h"
Include dependency graph for c_cut_polygon.h:
```



Functions

- void `c_getPlanePolygonFromReconstruction_RectangularCuboid_Polygon` (const `c_RectangularCuboid` *`a_rectangular_cuboid`, const `c_PlanarSeparator` *`a_separator`, const int *`a_plane_index`, `c_Polygon` *`a_polygon`)
Create a Polygon by truncating a Plane from a PlanarSeparator by a RectangularCuboid.
- void `c_getPlanePolygonFromReconstruction_RectangularCuboid_DividedPolygon` (const `c_RectangularCuboid` *`a_rectangular_cuboid`, const `c_PlanarSeparator` *`a_separator`, const int *`a_plane_index`, `c_DividedPolygon` *`a_divided_polygon`)
Create a DividedPolygon by truncating a Plane from a PlanarSeparator by a RectangularCuboid.
- double `c_getReconstructionSurfaceArea_RectangularCuboid` (const `c_RectangularCuboid` *`a_rectangular_cuboid`, const `c_PlanarSeparator` *`a_separator`)
Creates the interface polygons for the planes in a_separator that exist in a_rectangular_cuboid and returns the total area of these polygons.

7.2.1 Detailed Description

These C-style functions are mapped to functions available in src/cut_polygon.h.

This file maps to functions that deal with intersecting polygons with planes and calculating surface area from Polygons. This mostly means the creation of Polygons from intersections of Planes and Polyhedra, or intersection of Polygons with Planes to generate new (truncated) Polygons.

Individual documentation for each function is given alongside the function.

7.2.2 Function Documentation

7.2.2.1 `c_getPlanePolygonFromReconstruction_RectangularCuboid_DividedPolygon()`

```
void c_getPlanePolygonFromReconstruction_RectangularCuboid_DividedPolygon (
    const c_RectangularCuboid * a_rectangular_cuboid,
    const c_PlanarSeparator * a_separator,
    const int * a_plane_index,
    c_DividedPolygon * a_divided_polygon )
```

Create a DividedPolygon by truncating a Plane from a PlanarSeparator by a RectangularCuboid.

This function intersects the `a_plane_index` Plane of `a_separator` with a `a_rectangular_cuboid` in order to generate a `DividedPolygon`. If the `PlanarSeparator` consists of multiple planes, the `Polygon` object will also be intersected with the other planes in the `PlanarSeparator`. The centroid for the `DividedPolygon` is also updated before the function returns.

Parameters

in	<code>a_rectangular_cuboid</code>	Pointer to <code>RectangularCuboid</code> object that will be used to truncate the plane.
in	<code>a_separator</code>	Pointer to <code>PlanarSeparator</code> object that the plane from which a <code>DividedPolygon</code> is being created is taken.
in	<code>a_plane_index</code>	Index of plane in <code>a_separator</code> that the <code>DividedPolygon</code> will be created from.
out	<code>a_polygon</code>	Pointer to <code>DividedPolygon</code> object where the created <code>Polygon</code> will be stored.

7.2.2.2 c_getPlanePolygonFromReconstruction_RectangularCuboid_Polygon()

```
void c_getPlanePolygonFromReconstruction_RectangularCuboid_Polygon (
    const c_RectangularCuboid * a_rectangular_cuboid,
    const c_PlanarSeparator * a_separator,
    const int * a_plane_index,
    c_Polygon * a_polygon )
```

Create a Polygon by truncating a Plane from a PlanarSeparator by a RectangularCuboid.

This function intersects the `a_plane_index` Plane of `a_separator` with a `a_rectangular_cuboid` in order to generate a Polygon. If the PlanarSeparator consists of multiple planes, the Polygon object will also be intersected with the other planes in the PlanarSeparator.

Parameters

in	<code>a_rectangular_cuboid</code>	Pointer to RectangularCuboid object that will be used to truncate the plane.
in	<code>a_separator</code>	Pointer to PlanarSeparator object that the plane from which a Polygon is being created is taken.
in	<code>a_plane_index</code>	Index of plane in <code>a_separator</code> that the Polygon will be created from.
out	<code>a_polygon</code>	Pointer to Polygon object where the created Polygon will be stored.

7.2.2.3 c_getReconstructionSurfaceArea_RectangularCuboid()

```
double c_getReconstructionSurfaceArea_RectangularCuboid (
    const c_RectangularCuboid * a_rectangular_cuboid,
    const c_PlanarSeparator * a_separator )
```

Creates the interface polygons for the planes in `a_separator` that exist in `a_rectangular_cuboid` and returns the total area of these polygons.

This function creates Polygon objects for each plane in `a_separator` that exists solely in `a_rectangular_cuboid`. The area of these Polygon objects is summed and returned from the function. If `a_separator` contains multiple planes the polygons will also be intersected and truncated by them prior to the calculation of the surface area.

Parameters

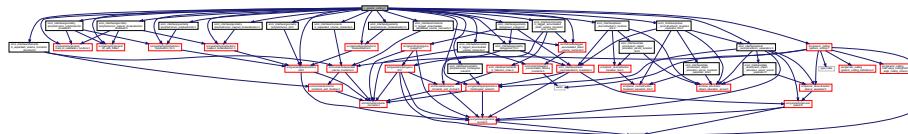
in	<code>a_rectangular_cuboid</code>	Pointer to RectangularCuboid object that will be used to truncate the planes in <code>a_separator</code> .
in	<code>a_separator</code>	Pointer to PlanarSeparator object from which the planes will be taken.

7.3 c_generic_cutting.h File Reference

```
#include "src/c_interface/geometry/polygons/c_polygon.h"
```

```
#include "src/c_interface/geometry/polygons/c_tri.h"
#include "src/c_interface/geometry/polyhedrons/c_capped_dodecahedron.h"
#include "src/c_interface/geometry/polyhedrons/c_capped_dodecahedron_doubles3.h"
#include "src/c_interface/geometry/polyhedrons/c_dodecahedron.h"
#include "src/c_interface/geometry/polyhedrons/c_polyhedron24.h"
#include "src/c_interface/geometry/polyhedrons/c_polyhedron24_doubles3.h"
#include "src/c_interface/geometry/polyhedrons/c_rectangular_cuboid.h"
#include "src/c_interface/geometry/polyhedrons/c_tet.h"
#include "src/c_interface/moments/c_separated_volume_moments.h"
#include "src/c_interface/moments/c_separated_volume_moments_doubles3.h"
#include "src/c_interface/moments/c_tagged_accumulated_listed_volume_moments_and_normal.h"
#include "src/c_interface/moments/c_tagged_accumulated_separated_volume_moments.h"
#include "src/c_interface/moments/c_tagged_accumulated_volume_moments.h"
#include "src/c_interface/planar_reconstruction/c_localized_separator_link.h"
#include "src/c_interface/planar_reconstruction/c_localizer_link.h"
#include "src/c_interface/planar_reconstruction/c_separators.h"
#include "src/generic_cutting/generic_cutting.h"
#include "src/geometry/general/new_pt_calculation_functors.h"
#include "src/geometry/general/pt_with_data.h"
#include "src/geometry/polyhedrons/capped_dodecahedron.h"
#include "src/geometry/polyhedrons/dodecahedron.h"
#include "src/geometry/polyhedrons/polyhedron_24.h"
#include "src/geometry/polyhedrons/rectangular_cuboid.h"
#include "src/geometry/polyhedrons/tet.h"
#include "src/moments/tagged_accumulated_listed_volume_moments.h"

Include dependency graph for c_generic_cutting.h:
```



Functions

- void [c_getVolumeMoments_setMethod](#) (const int *a_cutting_method)

Function to set the method used for cutting when a c_getNormalizedVolumeMoments function is called.
- void [c_getNormalizedVolumeMoments_Dodecahedron_By_LocalizedSeparatorLink_For_SeparatedVolumeMoments](#) (const [c_Dodecahedron](#) *a_dodecahedron, const [c_LocalizedSeparatorLink](#) *a_localized_separator_link, [c_SepVM](#) *a_moments_to_return)
- void [c_getNormalizedVolumeMoments_CappedDodecahedron_By_LocalizedSeparatorLink_For_SeparatedVolumeMoments](#) (const [c_CappedDodecahedron](#) *a_capped_dodecahedron, const [c_LocalizedSeparatorLink](#) *a_localized_separator_link, [c_SepVM](#) *a_moments_to_return)
- void [c_getNormalizedVolumeMoments_CappedDodecahedron_doubles3_By_LocalizedSeparatorLink_For_SeparatedVolumeMomentsAndDoubles3](#) (const [c_CappedDodecahedron_doubles3](#) *a_capped_dodecahedron, const [c_LocalizedSeparatorLink](#) *a_localized_separator_link, [c_SepVM_doubles3](#) *a_moments_to_return)
- void [c_getNormalizedVolumeMoments_Polyhedron24_By_LocalizedSeparatorLink_For_SeparatedVolumeMoments](#) (const [c_Polyhedron24](#) *a_polyhedron_24, const [c_LocalizedSeparatorLink](#) *a_localized_separator_link, [c_SepVM](#) *a_moments_to_return)

- void `c_getNormalizedVolumeMoments_Polyhedron24WithDoubles3_By_LocalizedSeparatorLink_For_SeparatedVolumeMomentsAndDoubles3` (const `c_Polyhedron24_doubles3` **a_polyhedron_24*, const `c_LocalizedSeparatorLink` **a_localized_separator_link*, `c_SepVM_doubles3` **a_moments_to_return*)
- void `c_getVolumeMoments_CappedDodecahedron_By_LocalizedSeparatorLink_For_SeparatedVolumeMoments` (const `c_CappedDodecahedron` **a_capped_dodecahedron*, const `c_LocalizedSeparatorLink` **a_localized_separator_link*, `c_SepVM` **a_moments_to_return*)
- void `c_getVolumeMoments_Dodecahedron_By_LocalizedSeparatorLink_For_SeparatedVolumeMoments` (const `c_Dodecahedron` **a_dodecahedron*, const `c_LocalizedSeparatorLink` **a_localized_separator_link*, `c_SepVM` **a_moments_to_return*)
- void `c_getVolumeMoments_Polyhedron24_By_LocalizedSeparatorLink_For_SeparatedVolumeMoments` (const `c_Polyhedron24` **a_polyhedron_24*, const `c_LocalizedSeparatorLink` **a_localized_separator_link*, `c_SepVM` **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_Tet_By_LocalizedSeparatorLink_For_SeparatedVolumeMoments` (const `c_Tet` **a_tet*, const `c_LocalizedSeparatorLink` **a_localized_separator_link*, `c_SepVM` **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_RectangularCuboid_By_PlanarSeparator_For_Volume` (const `c_RectangularCuboid` **a_rectangular_cuboid*, const `c_PlanarSeparator` **a_planar_separator*, double **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_Dodecahedron_By_PlanarSeparator_For_SeparatedVolumeMoments` (const `c_Dodecahedron` **a_dodecahedron*, const `c_PlanarSeparator` **a_planar_separator*, `c_SepVM` **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_Dodecahedron_By_LocalizedSeparator_For_SeparatedVolumeMoments` (const `c_Dodecahedron` **a_dodecahedron*, const `IRL::LocalizedSeparator` **a_localized_separator*, `c_SepVM` **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_CappedDodecahedron_By_LocalizedSeparatorLink_For_TaggedAccumulatedVolumeMoments_SeparatedVolumeMoments` (const `c_CappedDodecahedron` **a_capped_dodecahedron*, const `c_LocalizedSeparatorLink` **a_localized_separator_link*, `c_Tagged_AccumVM_SepVM` **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_Dodecahedron_By_LocalizedSeparatorLink_For_TaggedAccumulatedVolumeMoments_SeparatedVolumeMoments` (const `c_Dodecahedron` **a_dodecahedron*, const `c_LocalizedSeparatorLink` **a_localized_separator_link*, `c_Tagged_AccumVM_SepVM` **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_RectangularCuboid_By_PlanarSeparator_For_SeparatedVolumeMoments` (const `c_RectangularCuboid` **a_rectangular_cuboid*, const `c_PlanarSeparator` **a_planar_separator*, `c_SepVM` **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_Tri_By_LocalizerLink_For_TaggedAccumulatedVolumeMoments_VolumeMoments` (const `c_Tri` **a_tri*, const `c_LocalizerLink` **a_localizer_link*, `c_Tagged_AccumVM_VM` **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_Tri_By_PlanarLocalizer_For_Volume` (const `c_Tri` **a_tri*, const `c_PlanarLocalizer` **a_planar_localizer*, double **a_moments_to_return*)
- void `c_getNormalizedVolumeMoments_Polygon_By_PlanarLocalizer_For_Volume` (const `c_Polygon` **a_poly*, const `c_PlanarLocalizer` **a_planar_localizer*, double **a_moments_to_return*)
- void `c_getVolumeMoments_Tri_By_LocalizerLink_For_TaggedAccumulatedListedVolumeMoments_VolumeMomentsAndNormal` (const `c_Tri` **a_tri*, const `c_LocalizerLink` **a_localizer_link*, `c_Tagged_AccumListedVM_VMAN` **a_moments_to_return*)

7.3.1 Detailed Description

These C-style functions are mapped to functions available in `src/generic_cutting.h`.

This file deals with functions that compute volume moments for polyhedra and subdivided polyhedra. In principle, the first argument to the function is a pointer to a known polytope class available in `IRL`, such as a `Polygon`, `Tet`, or a `Dodecahedron`. The second argument is a pointer to a `PlanarSeparator`, `PlanarLocalizer`, `LocalizedSeparator`, or `LocalizedSeparatorLink` that will subdivide or otherwise restrict the integration area when calculating the volumetric moments. The third argument is a pointer to an object of the type of `VolumeMoments` that will be returned.

Individual documentation for each function is given alongside the function.

7.3.2 Function Documentation

7.3.2.1 c_getVolumeMoments_setMethod()

```
void c_getVolumeMoments_setMethod (
    const int * a_cutting_method )
```

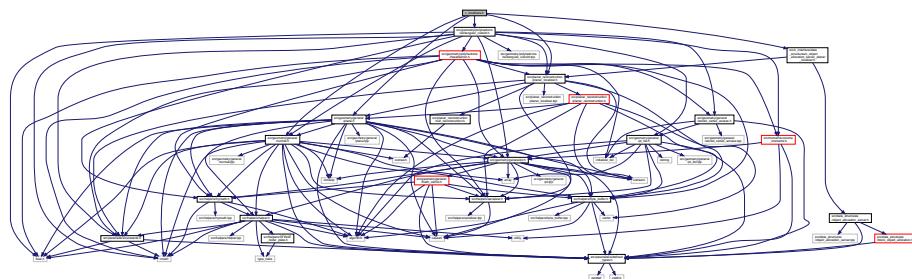
Function to set the method used for cutting when a c_getNormalizedVolumeMoments function is called.

- 0 : RecursiveSimplexCutting
- 1 : HalfEdgeCutting
- 2 : SimplexCutting

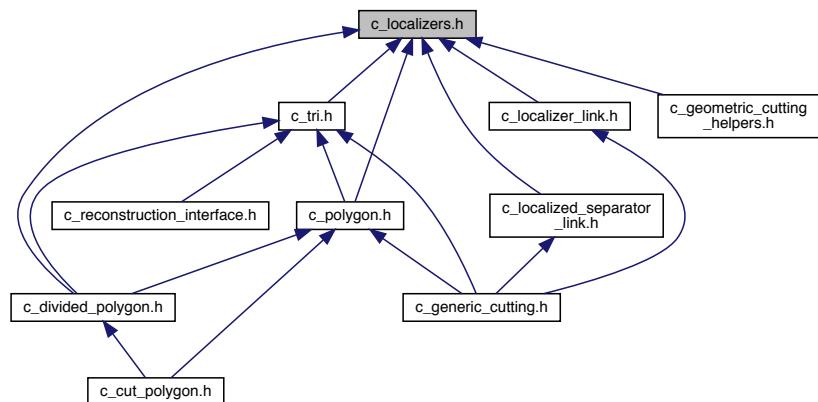
7.4 c_localizers.h File Reference

```
#include "src/c_interface/data_structures/c_object_allocation_server_planar-
_localizer.h"
#include "src/geometry/general/normal.h"
#include "src/geometry/general/plane.h"
#include "src/geometry/polyhedrons/rectangular_cuboid.h"
#include "src/planar_reconstruction/planar_localizer.h"
```

Include dependency graph for c_localizers.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct `c_PlanarLocalizer`

Functions

- void `c_PlanarLocalizer_new` (`c_PlanarLocalizer` *`a_self`)
- void `c_PlanarLocalizer_newFromObjectAllocationServer` (`c_PlanarLocalizer` *`a_self`, `c_ObjectAllocationServer_PlanarLocalizer` *`a_object_allocation_server`)
- void `c_PlanarLocalizer_delete` (`c_PlanarLocalizer` *`a_self`)
- void `c_PlanarLocalizer_addPlane` (`c_PlanarLocalizer` *`a_self`, const double *`a_normal`, const double *`a_distance`)
- void `c_PlanarLocalizer_setNumberOfPlanes` (`c_PlanarLocalizer` *`a_self`, const int *`a_number_to_set`)
- void `c_PlanarLocalizer_setPlane` (`c_PlanarLocalizer` *`a_self`, const int *`a_plane_index_to_set`, const double *`a_normal`, const double *`a_distance`)
- void `c_PlanarLocalizer_setFromRectangularCuboid` (`c_PlanarLocalizer` *`a_self`, const double *`a_lower_pt`, const double *`a_upper_pt`)
- void `c_PlanarLocalizer_printToScreen` (const `c_PlanarLocalizer` *`a_self`)

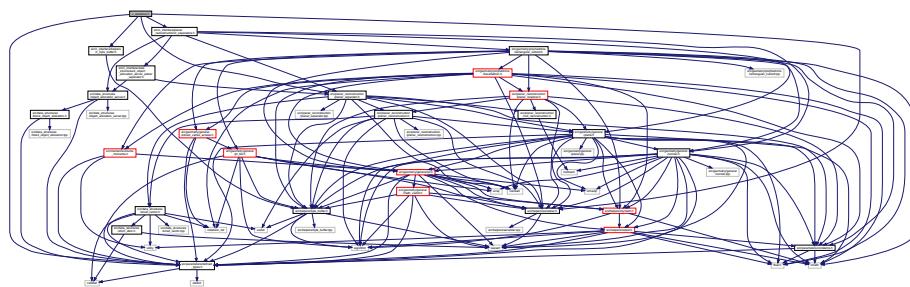
7.4.1 Detailed Description

These C-style functions are mapped to functions available in `src/reconstruction_interface.h`.

This file includes functions to place PlanarSeparator objects in geometries. These methods differ in what they require. For the individual needs of each reconstruction method, it is best to consult its specific documentation.

7.5 c_serializer.h File Reference

```
#include "src/c_interface/helpers/c_byte_buffer.h"
#include "src/c_interface/planar_reconstruction/c_separators.h"
#include "src/helpers/serializer.h"
#include "src/parameters/defined_types.h"
#include "src/planar_reconstruction/planar_separator.h"
Include dependency graph for c_serializer.h:
```



Functions

- void `c_Serializer_serializeAndPack_PlanarSeparator_ByteBuffer` (const `c_PlanarSeparator` *`a_separator`, `c_ByteBuffer` *`a_container`)
- void `c_Serializer_unpackAndStore_PlanarSeparator_ByteBuffer` (`c_PlanarSeparator` *`a_separator`, `c_ByteBuffer` *`a_container`)

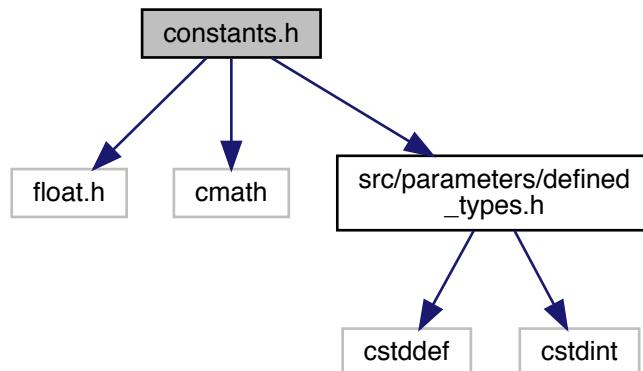
7.5.1 Detailed Description

These C-style functions are mapped to functions available in src/serializer.h.

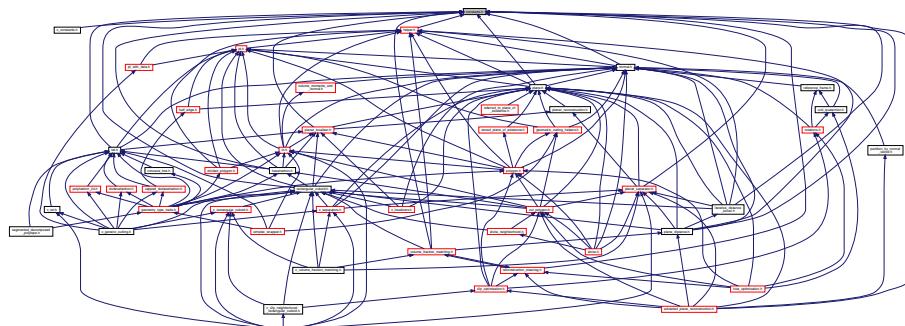
This file includes functions to handle the serialization and packing of IRL class objects into linear byte-buffers. This also includes the class ByteBuffer, which manages this linear packing and tracks its current buffer location, allowing easy sequential reading that takes place over multiple calls. These functions are mostly planned to be used along with MPI communication routines to send MPI_BYTETs between processors. This means that these functions assume a HOMOGENEOUS ARCHITECTURE, requiring all little-endian or all big-endian representation to be used.

7.6 constants.h File Reference

```
#include <float.h>
#include <cmath>
#include "src/parameters/defined_types.h"
Include dependency graph for constants.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- [IRL](#)

Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Functions

- void [IRL::setVolumeFractionBounds](#) (const double a_VF_low)
Set VF_LOW and VF_HIGH defined in src/constants.cpp.
- void [IRL::setVolumeFractionTolerance](#) (const double a_tolerance)
Set the volume fraction tolerance for iterative distance finding.
- void [IRL::setMinimumVolumeToTrack](#) (const double a_minimum_volume_to_track)
Function to set MINIMUM_VOLUME_TO_TRACK
- void [IRL::setMinimumSurfaceAreaToTrack](#) (const double a_minimum_surface_area_to_track)
Function to set MINIMUM_SURFACE_AREA_TO_TRACK

Variables

- static constexpr double [IRL::global_constants::ARBITRARILY_LARGE_DISTANCE](#) = 0.5 * DBL_MAX
An arbitrarily large distance for planes of single-phase cells.
- constexpr UnsignedIndex_t [IRL::global_constants::MAX_PLANAR_SEPARATOR_PLANES](#) = 3
Maximum number of planes that will below in a PlanarSeparator.
- constexpr UnsignedIndex_t [IRL::global_constants::MAX_PLANAR_LOCALIZER_PLANES](#) = 8
Maximum number of planes that will below in a PlanarLocalizer.
- constexpr UnsignedIndex_t [IRL::global_constants::MAX_PLANES_IN_OBJECT](#)
Max number of planes in a PlanarReconstruction based object.
- static constexpr double [IRL::global_constants::SAME_VEC](#) = 1.0 - 1.0e-4
Value that determines if two vectors are the same through $n_1 \cdot n_2 > SAMEVEC$.

7.6.1 Detailed Description

This file contains constants related to reconstructions and the computational geometry routines.

7.6.2 Variable Documentation

7.6.2.1 ARBITRARILY_LARGE_DISTANCE

```
constexpr double IRL::global_constants::ARBITRARILY_LARGE_DISTANCE = 0.5 * DBL_MAX [static]
```

An arbitrarily large distance for planes of single-phase cells.

An arbitrarily large distance for a plane of a single cell. Used so that no matter the normal, the cell will be seen as single phase once cut computationally. The only requirement on for ARBITRARILY_LARGE_DISTANCE is that it is larger than the size of the domain (i.e. $> Lx + Ly + Lz$) if global datum reconstructions are used. If only local reconstruction are used, it only needs to be large enough to place the place outside of the unit cell.

7.6.2.2 MAX_PLANES_IN_OBJECT

```
constexpr UnsignedIndex_t IRL::global_constants::MAX_PLANES_IN_OBJECT
```

Initial value:

```
= MAX_PLANAR_SEPARATOR_PLANES >      MAX_PLANAR_LOCALIZER_PLANES
? MAX_PLANAR_SEPARATOR_PLANES
: MAX_PLANAR_LOCALIZER_PLANES
```

Max number of planes in a PlanarReconstruction based object.

7.7 f_bytebuffer_class.f90 File Reference

This file contains the Fortran interface for the ByteBuffer class.

Data Types

- type [f_bytebuffer_class::c_bytebuffer](#)
- type [f_bytebuffer_class::bytebuffer_type](#)
- interface [f_bytebuffer_class::new](#)
- interface [f_bytebuffer_class::getcobject](#)
- interface [f_bytebuffer_class::getsize](#)
- interface [f_bytebuffer_class::setszie](#)
- interface [f_bytebuffer_class::resetbufferpointer](#)
- interface [f_bytebuffer_class::dataptr](#)
- interface [f_bytebuffer_class::F_ByteBuffer_new](#)
- interface [f_bytebuffer_class::F_ByteBuffer_delete](#)
- interface [f_bytebuffer_class::F_ByteBuffer_getSize](#)
- interface [f_bytebuffer_class::F_ByteBuffer_setSize](#)
- interface [f_bytebuffer_class::F_ByteBuffer_resetBufferPointer](#)
- interface [f_bytebuffer_class::F_ByteBuffer_dataPtr](#)

Modules

- module [f_bytebuffer_class](#)

A fortran type class that allows the creation of IRL's ByteBuffer class along with enabling some of its methods.

Functions/Subroutines

- impure elemental subroutine [f_bytebuffer_class::bytebuffer_class_delete](#) (this)
- subroutine [f_bytebuffer_class::bytebuffer_class_new](#) (this)
- type(c_bytebuffer) function [f_bytebuffer_class::bytebuffer_class_getcobject](#) (this)
- integer(irl_largeoffsetindex_t) function [f_bytebuffer_class::bytebuffer_class_getsize](#) (this)
- subroutine [f_bytebuffer_class::bytebuffer_class_setszie](#) (this, a_size)
- subroutine [f_bytebuffer_class::bytebuffer_class_resetbufferpointer](#) (this)
- integer(irl_byte_t) function, dimension(:), pointer [f_bytebuffer_class::bytebuffer_class_dataptr](#) (this)

7.7.1 Detailed Description

This file contains the Fortran interface for the ByteBuffer class.

7.8 f_cappeddodecahedron_class.f90 File Reference

This file contains the Fortran interface for the CappedDodecahedron class.

Data Types

- type [f_cappeddodecahedron_class::c_cappeddodecahedron](#)
- type [f_cappeddodecahedron_class::cappeddodecahedron_type](#)
- interface [f_cappeddodecahedron_class::new](#)
- interface [f_cappeddodecahedron_class::getobject](#)
- interface [f_cappeddodecahedron_class::construct](#)
- interface [f_cappeddodecahedron_class::adjustcaptomatchvolume](#)
- interface [f_cappeddodecahedron_class::getboundingpts](#)
- interface [f_cappeddodecahedron_class::getpt](#)
- interface [f_cappeddodecahedron_class::F_CappedDodecahedron_new](#)
- interface [f_cappeddodecahedron_class::F_CappedDodecahedron_delete](#)
- interface [f_cappeddodecahedron_class::F_CappedDodecahedron_construct](#)
- interface [f_cappeddodecahedron_class::F_CappedDodecahedron_adjustCapToMatchVolume](#)
- interface [f_cappeddodecahedron_class::F_CappedDodecahedron_getBoundingPts](#)
- interface [f_cappeddodecahedron_class::F_CappedDodecahedron_getPt](#)

Modules

- module [f_cappeddodecahedron_class](#)

A fortran type class that allows the creation of IRL's CappedDodecahedron class along with enabling some of its methods.

Functions/Subroutines

- impure elemental subroutine [f_cappeddodecahedron_class::cappeddodecahedron_class_delete](#) (this)
- subroutine [f_cappeddodecahedron_class::cappeddodecahedron_class_new](#) (this)
- type([c_cappeddodecahedron](#)) function [f_cappeddodecahedron_class::cappeddodecahedron_class_getobject](#) (this)
- subroutine [f_cappeddodecahedron_class::cappeddodecahedron_class_construct](#) (this, a_dodecahedron)
- subroutine [f_cappeddodecahedron_class::cappeddodecahedron_class_adjustcaptomatchvolume](#) (this, a_correct_signed_volume)
- subroutine [f_cappeddodecahedron_class::cappeddodecahedron_class_getboundingpts](#) (this, a_lower_pt, a_upper_pt)
- real([irl_double](#)) function, dimension(3) [f_cappeddodecahedron_class::cappeddodecahedron_class_getpt](#) (this, a_index)

7.8.1 Detailed Description

This file contains the Fortran interface for the CappedDodecahedron class.

7.9 f_cappeddodecahedron_doubles3_class.f90 File Reference

This file contains the Fortran interface for the CappedDodecahedron_doubles3 class.

Data Types

- type `f_cappeddodecahedron_doubles3_class::c_cappeddodecahedron_doubles3`
- type `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_type`
- interface `f_cappeddodecahedron_doubles3_class::new`
- interface `f_cappeddodecahedron_doubles3_class::getobject`
- interface `f_cappeddodecahedron_doubles3_class::construct`
- interface `f_cappeddodecahedron_doubles3_class::adjustcaptomatchvolume`
- interface `f_cappeddodecahedron_doubles3_class::getboundingpts`
- interface `f_cappeddodecahedron_doubles3_class::getpt`
- interface `f_cappeddodecahedron_doubles3_class::setpt`
- interface `f_cappeddodecahedron_doubles3_class::getdata`
- interface `f_cappeddodecahedron_doubles3_class::setdata`
- interface `f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_new`
- interface `f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_delete`
- interface `f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_construct`
- interface `f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_adjustCapToMatchVolume`
- interface `f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getBoundingPts`
- interface `f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getPt`
- interface `f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_setPt`
- interface `f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_getData`
- interface `f_cappeddodecahedron_doubles3_class::F_CappedDodecahedron_doubles3_setData`

Modules

- module `f_cappeddodecahedron_doubles3_class`

A fortran type class that allows the creation of IRL's CappedDodecahedron_doubles3 class along with enabling some of its methods.

Functions/Subroutines

- impure elemental subroutine `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_delete` (`this`)
- subroutine `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_new` (`this`)
- type(`c_cappeddodecahedron_doubles3`) function `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_getobject` (`this`)
- subroutine `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_construct` (`this`, `a_dodecahedron`, `a_attached_data`)
- subroutine `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_adjustcaptomatchvolume` (`this`, `a_correct_signed_volume`)
- subroutine `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_getboundingpts` (`this`, `a_lower_pt`, `a_upper_pt`)
- real(`irl_double`) function, dimension(3) `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_getpt` (`this`, `a_index`)
- subroutine `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_setpt` (`this`, `a_index`, `a_pt`)
- real(`irl_double`) function, dimension(3) `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_getdata` (`this`, `a_index`)
- subroutine `f_cappeddodecahedron_doubles3_class::cappeddodecahedron_doubles3_class_setdata` (`this`, `a_index`, `a_data`)

7.9.1 Detailed Description

This file contains the Fortran interface for the CappedDodecahedron_doubles3 class.

7.10 f_constants.f90 File Reference

This file contains the Fortran interface to [IRL](#) functions that deal with setting constants.

Data Types

- interface [f_constants::F_Constants_setVolumeFractionBounds](#)
- interface [f_constants::F_Constants_setVolumeFractionToleranceForDistanceFinding](#)
- interface [f_constants::F_Constants_setMinimumVolumeToTrack](#)
- interface [f_constants::F_Constants_setMinimumSurfaceAreaToTrack](#)

Modules

- module [f_constants](#)

This module contains mappings to the [IRL](#) C interface that deal with setting global constants that are used in the [IRL](#) library.

Functions/Subroutines

- subroutine [f_constants::constants_setvolumefractionbounds](#) (a_VF_low)
- subroutine [f_constants::constants_setvolumefractiontolerancefordistancefinding](#) (a_tolerance)
- subroutine [f_constants::constants_setminimumvolumetotrack](#) (a_minimum_volume_to_track)
- subroutine [f_constants::constants_setminimumsurfaceareatotrack](#) (a_minimum_surface_area_to_track)

7.10.1 Detailed Description

This file contains the Fortran interface to [IRL](#) functions that deal with setting constants.

7.11 f_cutmolygon.f90 File Reference

This file deals with intersecting polygons and generating polygons corresponding to planar reconstructions.

Data Types

- interface [f_cutmolygon::getplanepolygonfromreconstruction](#)
- interface [f_cutmolygon::getreconstructionsurfacearea](#)
- interface [f_cutmolygon::F_getPlanePolygonFromReconstruction_RC_Poly](#)
- interface [f_cutmolygon::F_getPlanePolygonFromReconstruction_RC_DivPoly](#)
- interface [f_cutmolygon::F_getReconstructionSurfaceArea_RC](#)

Modules

- module [f_cutpolygon](#)

This module contains mappings to the [IRL](#) C interface that deal with intersecting planes to generate polygons and creating polygons that are representative of planar reconstructions in given cells.

Functions/Subroutines

- subroutine [f_cutpolygon::getplanepolygonfromreconstruction_rc_poly](#) (a_rectangular_cuboid, a_<planar_separator, a_plane_index, a_polygon)
- subroutine [f_cutpolygon::getplanepolygonfromreconstruction_rc_divpoly](#) (a_rectangular_cuboid, a_<planar_separator, a_plane_index, a_divided_polygon)
- real(irl_double) function [f_cutpolygon::getreconstructionsurfacearea_rc](#) (a_rectangular_cuboid, a_<planar_separator)

7.11.1 Detailed Description

This file deals with intersecting polygons and generating polygons corresponding to planar reconstructions.

7.12 f_dividedpolygon_class.f90 File Reference

This file contains the Fortran interface for the DividedPolygon class.

Data Types

- type [f_dividedpolygon_class::c_dividedpolygon](#)
- type [f_dividedpolygon_class::dividedpolygon_type](#)
- interface [f_dividedpolygon_class::new](#)
- interface [f_dividedpolygon_class::getcobject](#)
- interface [f_dividedpolygon_class::construct](#)
- interface [f_dividedpolygon_class::constructfrompolygon](#)
- interface [f_dividedpolygon_class::resetcentroid](#)
- interface [f_dividedpolygon_class::getnumberofsimplicesindecomposition](#)
- interface [f_dividedpolygon_class::getsimplexfromdecomposition](#)
- interface [f_dividedpolygon_class::calculatenormal](#)
- interface [f_dividedpolygon_class::getlocalizer](#)
- interface [f_dividedpolygon_class::reverseptordering](#)
- interface [f_dividedpolygon_class::getboundingpts](#)
- interface [f_dividedpolygon_class::getnumberofvertices](#)
- interface [f_dividedpolygon_class::getpt](#)
- interface [f_dividedpolygon_class::zeropolygon](#)
- interface [f_dividedpolygon_class::calculatesurfacearea](#)
- interface [f_dividedpolygon_class::calculatesign](#)
- interface [f_dividedpolygon_class::setplaneofexistence](#)
- interface [f_dividedpolygon_class::calculateandsetplaneofexistence](#)
- interface [f_dividedpolygon_class::getplaneofexistence](#)
- interface [f_dividedpolygon_class::printtoscreen](#)
- interface [f_dividedpolygon_class::F_DividedPolygon_new](#)
- interface [f_dividedpolygon_class::F_DividedPolygon_delete](#)

- interface `f_dividedpolygon_class::F_DividedPolygon_construct`
- interface `f_dividedpolygon_class::F_DividedPolygon_constructFromPolygon`
- interface `f_dividedpolygon_class::F_DividedPolygon_resetCentroid`
- interface `f_dividedpolygon_class::F_DividedPolygon_getNumberOfSimplicesInDecomposition`
- interface `f_dividedpolygon_class::F_DividedPolygon_getSimplexFromDecomposition`
- interface `f_dividedpolygon_class::F_DividedPolygon_calculateNormal`
- interface `f_dividedpolygon_class::F_DividedPolygon_getLocalizer`
- interface `f_dividedpolygon_class::F_DividedPolygon_reversePtOrdering`
- interface `f_dividedpolygon_class::F_DividedPolygon_getBoundingPts`
- interface `f_dividedpolygon_class::F_DividedPolygon_getNumberOfPts`
- interface `f_dividedpolygon_class::F_DividedPolygon_getPt`
- interface `f_dividedpolygon_class::F_DividedPolygon_zeroPolygon`
- interface `f_dividedpolygon_class::F_DividedPolygon_calculateSurfaceArea`
- interface `f_dividedpolygon_class::F_DividedPolygon_calculateSign`
- interface `f_dividedpolygon_class::F_DividedPolygon_setPlaneOfExistence`
- interface `f_dividedpolygon_class::F_DividedPolygon_calculateAndSetPlaneOfExistence`
- interface `f_dividedpolygon_class::F_DividedPolygon_getPlaneOfExistence`
- interface `f_dividedpolygon_class::F_DividedPolygon_printToScreen`

Modules

- module `f_dividedpolygon_class`

A fortran type class that allows the creation of IRL's DividedPolygon class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_dividedpolygon_class::dividedpolygon_class_new` (this)
- impure elemental subroutine `f_dividedpolygon_class::dividedpolygon_class_delete` (this)
- type(`c_dividedpolygon`) function `f_dividedpolygon_class::dividedpolygon_class_getobject` (this)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_construct` (this, a_npts, a_pts)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_constructfrompolygon` (this, a_polygon)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_resetcentroid` (this)
- integer(irl_unsignedindex_t) function `f_dividedpolygon_class::dividedpolygon_class_getnumberofsimplicesindecomposition` (this)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_getsimplexfromdecomposition` (this, a_\leftarrow tri_number_to_get, a_tri_in_decomposition)
- real(irl_double) function, dimension(1:3) `f_dividedpolygon_class::dividedpolygon_class_calculatenormal` (this)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_getlocalizer` (this, a_planar_localizer)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_reverseptordering` (this)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_getboundingpts` (this, a_lower_pt, a_\leftarrow upper_pt)
- integer(irl_unsignedindex_t) function `f_dividedpolygon_class::dividedpolygon_class_getnumberoftpts` (this)
- real(irl_double) function, dimension(3) `f_dividedpolygon_class::dividedpolygon_class_getpt` (this, a_\leftarrow index)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_zeropolygon` (this)
- real(irl_double) function `f_dividedpolygon_class::dividedpolygon_class_calculatesurfacearea` (this)
- real(irl_double) function `f_dividedpolygon_class::dividedpolygon_class_calculatesign` (this)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_setplaneofexistence` (this, a_plane)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_calculateandsetplaneofexistence` (this)
- real(irl_double) function, dimension(4) `f_dividedpolygon_class::dividedpolygon_class_getplaneofexistence` (this)
- subroutine `f_dividedpolygon_class::dividedpolygon_class_printtoscreen` (this)

7.12.1 Detailed Description

This file contains the Fortran interface for the DividedPolygon class.

7.13 f_dodecahedron_class.f90 File Reference

This file contains the Fortran interface for the Dodecahedron class.

Data Types

- type `f_dodecahedron_class::c_dodecahedron`
- type `f_dodecahedron_class::dodecahedron_type`
- interface `f_dodecahedron_class::new`
- interface `f_dodecahedron_class::getcobject`
- interface `f_dodecahedron_class::construct`
- interface `f_dodecahedron_class::getboundingpts`
- interface `f_dodecahedron_class::F_Dodecahedron_new`
- interface `f_dodecahedron_class::F_Dodecahedron_delete`
- interface `f_dodecahedron_class::F_Dodecahedron_construct`
- interface `f_dodecahedron_class::F_Dodecahedron_getBoundingPts`

Modules

- module `f_dodecahedron_class`

A fortran type class that allows the creation of IRL's Dodecahedron class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_dodecahedron_class::dodecahedron_class_new` (this)
- impure elemental subroutine `f_dodecahedron_class::dodecahedron_class_delete` (this)
- type(`c_dodecahedron`) function `f_dodecahedron_class::dodecahedron_class_getcobject` (this)
- subroutine `f_dodecahedron_class::dodecahedron_class_construct` (this, a_transported_cell)
- subroutine `f_dodecahedron_class::dodecahedron_class_getboundingpts` (this, a_lower_pt, a_upper_pt)

7.13.1 Detailed Description

This file contains the Fortran interface for the Dodecahedron class.

7.14 f_geometriccuttinghelpers.f90 File Reference

This file provides access to helper functions often used during geometric cutting.

Data Types

- interface `f_geometriccuttinghelpers::isptinternal`
- interface `f_geometriccuttinghelpers::F_isPtInternal_PS`
- interface `f_geometriccuttinghelpers::F_isPtInternal_PL`

Modules

- module `f_geometriccuttinghelpers`

This module contains mappings to the [IRL](#) C interface that provides access to functions often used to geometric cutting operations. See the C interface file `src/c_interface/c_geometric_cutting_helpers.h` for more information.

Functions/Subroutines

- logical(1) function `f_geometriccuttinghelpers::isptinternal_ps` (`a_pt, a_separator`)
- logical(1) function `f_geometriccuttinghelpers::isptinternal_pl` (`a_pt, a_localizer`)

7.14.1 Detailed Description

This file provides access to helper functions often used during geometric cutting.

7.15 f_getvolumemoments.f90 File Reference

This file deals with subdividing and integrating volume moments for polyhedra.

Data Types

- interface `f_getvolumemoments::getvolumemoments_setmethod`
- interface `f_getvolumemoments::getnormalizedvolumemoments`
- interface `f_getvolumemoments::getvolumemoments`
- interface `f_getvolumemoments::F_GVM_setMethod`
- interface `f_getvolumemoments::F_GNVM_D_By_LSL_For_SVM`
- interface `f_getvolumemoments::F_GNVM_CD_By_LSL_For_SVM`
- interface `f_getvolumemoments::F_GNVM_CDWD3_By_LSL_For_SVMAD3`
- interface `f_getvolumemoments::F_GNVM_P24_By_LSL_For_SVM`
- interface `f_getvolumemoments::F_GNVM_P24WD3_By_LSL_For_SVMAD3`
- interface `f_getvolumemoments::F_GVM_CD_By_LSL_For_SVM`
- interface `f_getvolumemoments::F_GVM_D_By_LSL_For_SVM`
- interface `f_getvolumemoments::F_GVM_P24_By_LSL_For_SVM`
- interface `f_getvolumemoments::F_GNVM_Tet_By_LSL_For_SVM`
- interface `f_getvolumemoments::F_GNVM_RC_By_PS_For_V`
- interface `f_getvolumemoments::F_GNVM_D_By_PS_For_SVM`
- interface `f_getvolumemoments::F_GNVM_CD_By_LSL_For_TagAccumVM_SVM`
- interface `f_getvolumemoments::F_GNVM_D_By_LSL_For_TagAccumVM_SVM`
- interface `f_getvolumemoments::F_GNVM_RC_By_PS_For_SVM`
- interface `f_getvolumemoments::F_GNVM_Tri_By_LL_For_TagAVM_VM`
- interface `f_getvolumemoments::F_GNVM_Tri_By_PL_For_V`
- interface `f_getvolumemoments::F_GNVM_Poly_By_PL_For_V`
- interface `f_getvolumemoments::F_GVM_Tri_By_LL_For_TagALVM_VMAN`

Modules

- module **f_getvolumemoments**

This module contains mappings to the [IRL](#) C interface that deal with intersecting polyhedron volumes and integrating these volumes to obtain volumetric moments.

Functions/Subroutines

- subroutine **f_getvolumemoments::gvm_setmethod** (a_cutting_method)
- subroutine **f_getvolumemoments::gnvm_d_by_lsl_for_svm** (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_cd_by_lsl_for_svm** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_cdwd3_by_lsl_for_svmad3** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_p24_by_lsl_for_svm** (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_p24wd3_by_lsl_for_svmad3** (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gvm_cd_by_lsl_for_svm** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gvm_d_by_lsl_for_svm** (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gvm_p24_by_lsl_for_svm** (a_polyhedron_24, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_tet_by_lsl_for_svm** (a_tet, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_rc_by_ps_for_v** (a_rectangular_cuboid, a_planar_separator, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_d_by_ps_for_svm** (a_Dodecahedron, a_planar_separator, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_cd_by_lsl_for_tagaccumvm_svm** (a_Capped_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_d_by_lsl_for_tagaccumvm_svm** (a_Dodecahedron, a_localized_separator_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_rc_by_ps_for_svm** (a_rectangular_cuboid, a_planar_separator, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_tri_by_ll_for_tagavm_vm** (a_tri, a_localizer_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_tri_by_ll_for_tagavm_vm** (a_tri, a_localizer_link, a_moments_to_return)
- subroutine **f_getvolumemoments::gnvm_poly_by_pl_for_v** (a_polygon, a_planar_localizer, a_moments_to_return)
- subroutine **f_getvolumemoments::gvm_tri_by_ll_for_tagalvm_vman** (a_tri, a_localizer_link, a_moments_to_return)

7.15.1 Detailed Description

This file deals with subdividing and integrating volume moments for polyhedra.

7.16 f_localizedseparatorlink_class.f90 File Reference

This file allows use of the [IRL](#) LocalizedSeparatorLink class through a fortran interface.

Data Types

- type `f_localizedseparatorlink_class::c_localizedseparatorlink`
- type `f_localizedseparatorlink_class::localizedseparatorlink_type`
- interface `f_localizedseparatorlink_class::new`
- interface `f_localizedseparatorlink_class::getcobject`
- interface `f_localizedseparatorlink_class::setid`
- interface `f_localizedseparatorlink_class::getid`
- interface `f_localizedseparatorlink_class::setedgeconnectivity`
- interface `f_localizedseparatorlink_class::setedgeconnectivitynull`
- interface `f_localizedseparatorlink_class::F_LocalizedSeparatorLink_new`
- interface `f_localizedseparatorlink_class::F_LocalizedSeparatorLink_newFromObjectAllocationServer`
- interface `f_localizedseparatorlink_class::F_LocalizedSeparatorLink_delete`
- interface `f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setId`
- interface `f_localizedseparatorlink_class::F_LocalizedSeparatorLink_getId`
- interface `f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivity`
- interface `f_localizedseparatorlink_class::F_LocalizedSeparatorLink_setEdgeConnectivityNull`

Modules

- module `f_localizedseparatorlink_class`

A fortran type class that allows the creation of [IRL](#)'s LocalizedSeparatorLink class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_localizedseparatorlink_class::localizedseparatorlink_class_new` (`this`, `a_planar_localizer`, `a_planar_separator`)
- subroutine `f_localizedseparatorlink_class::localizedseparatorlink_class_newfromobjectallocationserver` (`this`, `a_object_allocation_server`, `a_planar_localizer`, `a_planar_separator`)
- impure elemental subroutine `f_localizedseparatorlink_class::localizedseparatorlink_class_delete` (`this`)
- type(`c_localizedseparatorlink`) function `f_localizedseparatorlink_class::localizedseparatorlink_class_getcobject` (`this`)
- subroutine `f_localizedseparatorlink_class::localizedseparatorlink_class_setid` (`this`, `a_id`)
- integer(`irl_unsignedindex_t`) function `f_localizedseparatorlink_class::localizedseparatorlink_class_getid` (`this`)
- subroutine `f_localizedseparatorlink_class::localizedseparatorlink_class_setedgeconnectivity` (`this`, `a_plane_index`, `a_neighboring_LocalizedSeparatorLink`)
- subroutine `f_localizedseparatorlink_class::localizedseparatorlink_class_setedgeconnectivitynull` (`this`, `a_plane_index`)

7.16.1 Detailed Description

This file allows use of the [IRL](#) LocalizedSeparatorLink class through a fortran interface.

7.17 f_localizerlink_class.f90 File Reference

This file allows use of the [IRL](#) LocalizerLink class through a fortran interface.

Data Types

- type `f_localizerlink_class::c_localizerlink`
- type `f_localizerlink_class::localizerlink_type`
- interface `f_localizerlink_class::new`
- interface `f_localizerlink_class::getobject`
- interface `f_localizerlink_class::setid`
- interface `f_localizerlink_class::getid`
- interface `f_localizerlink_class::setedgeconnectivity`
- interface `f_localizerlink_class::setedgeconnectivitynull`
- interface `f_localizerlink_class::F_LocalizerLink_new`
- interface `f_localizerlink_class::F_LocalizerLink_newFromObjectAllocationServer`
- interface `f_localizerlink_class::F_LocalizerLink_delete`
- interface `f_localizerlink_class::F_LocalizerLink_setId`
- interface `f_localizerlink_class::F_LocalizerLink_getId`
- interface `f_localizerlink_class::F_LocalizerLink_setEdgeConnectivity`
- interface `f_localizerlink_class::F_LocalizerLink_setEdgeConnectivityNull`

Modules

- module `f_localizerlink_class`

A fortran type class that allows the creation of [IRL](#)'s LocalizerLink class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_localizerlink_class::localizerlink_class_new` (this, a_planar_localizer)
- subroutine `f_localizerlink_class::localizerlink_class_newfromobjectallocationserver` (this, a_object_allocation_server, a_planar_localizer)
- impure elemental subroutine `f_localizerlink_class::localizerlink_class_delete` (this)
- type(`c_localizerlink`) function `f_localizerlink_class::localizerlink_class_getobject` (this)
- subroutine `f_localizerlink_class::localizerlink_class_setid` (this, a_id)
- integer(`irl_unsignedindex_t`) function `f_localizerlink_class::localizerlink_class_getid` (this)
- subroutine `f_localizerlink_class::localizerlink_class_setedgeconnectivity` (this, a_plane_index, a_neighboring_LocalizerLink)
- subroutine `f_localizerlink_class::localizerlink_class_setedgeconnectivitynull` (this, a_plane_index)

7.17.1 Detailed Description

This file allows use of the [IRL](#) LocalizerLink class through a fortran interface.

7.18 f_objectallocationserver_localizedseparatorlink_class.f90 File Reference

This file allows use of the [IRL](#) ObjectAllocationServer<LocalizedSeparatorLink> class through a fortran interface.

Data Types

- type [f_objectallocationserver_localizedseparatorlink_class::c_objectallocationserver_localizedseparatorlink](#)
- type [f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_type](#)
- interface [f_objectallocationserver_localizedseparatorlink_class::new](#)
- interface [f_objectallocationserver_localizedseparatorlink_class::getcobject](#)
- interface [f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparatorLink_new](#)
- interface [f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparatorLink_delete](#)

Modules

- module [f_objectallocationserver_localizedseparatorlink_class](#)

A fortran type class that allows the creation of [IRL's ObjectAllocationServer<LocalizedSeparatorLink>](#) class along with enabling some of its methods.

Functions/Subroutines

- subroutine [f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_class_new](#) (this, a_number_to_allocate)
- impure elemental subroutine [f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_class_delete](#) (this)
- type([c_objectallocationserver_localizedseparatorlink](#)) function [f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_class_getcobject](#) (this)

7.18.1 Detailed Description

This file allows use of the [IRL ObjectAllocationServer<LocalizedSeparatorLink>](#) class through a fortran interface.

7.19 f_objectallocationserver_localizerlink_class.f90 File Reference

This file allows use of the [IRL ObjectAllocationServer<LocalizerLink>](#) class through a fortran interface.

Data Types

- type [f_objectallocationserver_localizerlink_class::c_objectallocationserver_localizerlink](#)
- type [f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_type](#)
- interface [f_objectallocationserver_localizerlink_class::new](#)
- interface [f_objectallocationserver_localizerlink_class::getcobject](#)
- interface [f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_new](#)
- interface [f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_delete](#)

Modules

- module [f_objectallocationserver_localizerlink_class](#)

A fortran type class that allows the creation of [IRL's ObjectAllocationServer<LocalizerLink>](#) class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_class::new` (this, a_number_to_allocate)
- impure elemental subroutine `f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_class_delete` (this)
- type(c_objectallocationserver_localizerlink) function `f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_class_getcobject` (this)

7.19.1 Detailed Description

This file allows use of the [IRL ObjectAllocationServer<LocalizerLink>](#) class through a fortran interface.

7.20 f_objectallocationserver_planarlocalizer_class.f90 File Reference

This file allows use of the [IRL ObjectAllocationServer<PlanarLocalizer>](#) class through a fortran interface.

Data Types

- type `f_objectallocationserver_planarlocalizer_class::c_objectallocationserver_planarlocalizer`
- type `f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer_type`
- interface `f_objectallocationserver_planarlocalizer_class::new`
- interface `f_objectallocationserver_planarlocalizer_class::getcobject`
- interface `f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_new`
- interface `f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_delete`

Modules

- module `f_objectallocationserver_planarlocalizer_class`

A fortran type class that allows the creation of [IRL's ObjectAllocationServer<PlanarLocalizer>](#) class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer::class_new` (this, a_number_to_allocate)
- impure elemental subroutine `f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer_class_delete` (this)
- type(c_objectallocationserver_planarlocalizer) function `f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer_class_getcobject` (this)

7.20.1 Detailed Description

This file allows use of the [IRL ObjectAllocationServer<PlanarLocalizer>](#) class through a fortran interface.

7.21 f_objectallocationserver_planarseparator_class.f90 File Reference

This file allows use of the [IRL](#) ObjectAllocationServer<PlanarSeparator> class through a fortran interface.

Data Types

- type [f_objectallocationserver_planarseparator_class::c_objectallocationserver_planarseparator](#)
- type [f_objectallocationserver_planarseparator_class::objectallocationserver_planarseparator_type](#)
- interface [f_objectallocationserver_planarseparator_class::new](#)
- interface [f_objectallocationserver_planarseparator_class::getcobject](#)
- interface [f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_PlanarSeparator_new](#)
- interface [f_objectallocationserver_planarseparator_class::F_ObjectAllocationServer_PlanarSeparator_delete](#)

Modules

- module [f_objectallocationserver_planarseparator_class](#)

A fortran type class that allows the creation of [IRL](#)'s ObjectAllocationServer<PlanarSeparator> class along with enabling some of its methods.

Functions/Subroutines

- subroutine [f_objectallocationserver_planarseparator_class::objectallocationserver_planarseparator_class_new](#) (this, a_number_to_allocate)
- impure elemental subroutine [f_objectallocationserver_planarseparator_class::objectallocationserver_planarseparator_class_delete](#) (this)
- type([c_objectallocationserver_planarseparator](#)) function [f_objectallocationserver_planarseparator_class::objectallocationserver_planarseparator_class_getcobject](#) (this)

7.21.1 Detailed Description

This file allows use of the [IRL](#) ObjectAllocationServer<PlanarSeparator> class through a fortran interface.

7.22 f_planarlocalizer_class.f90 File Reference

This file allows use of the [IRL](#) PlanarLocalizer class through a fortran interface.

Data Types

- type [f_planarlocalizer_class::c_planarlocalizer](#)
- type [f_planarlocalizer_class::planarlocalizer_type](#)
- interface [f_planarlocalizer_class::new](#)
- interface [f_planarlocalizer_class::getcobject](#)
- interface [f_planarlocalizer_class::addplane](#)
- interface [f_planarlocalizer_class::setnumberofplanes](#)
- interface [f_planarlocalizer_class::setplane](#)
- interface [f_planarlocalizer_class::setfromrectangularcuboid](#)
- interface [f_planarlocalizer_class::printtoscreen](#)
- interface [f_planarlocalizer_class::F_PlanarLocalizer_new](#)
- interface [f_planarlocalizer_class::F_PlanarLocalizer_newFromObjectAllocationServer](#)
- interface [f_planarlocalizer_class::F_PlanarLocalizer_delete](#)
- interface [f_planarlocalizer_class::F_PlanarLocalizer_addPlane](#)
- interface [f_planarlocalizer_class::F_PlanarLocalizer_setNumberOfPlanes](#)
- interface [f_planarlocalizer_class::F_PlanarLocalizer_setPlane](#)
- interface [f_planarlocalizer_class::F_PlanarLocalizer_setFromRectangularCuboid](#)
- interface [f_planarlocalizer_class::F_PlanarLocalizer_printToScreen](#)

Modules

- module [f_planarlocalizer_class](#)

A fortran type class that allows the creation of [IRL](#)'s PlanarLocalizer class along with enabling some of its methods.

Functions/Subroutines

- subroutine [f_planarlocalizer_class::planarlocalizer_class_new](#) (this)
- subroutine [f_planarlocalizer_class::planarlocalizer_class_newfromobjectallocationserver](#) (this, a_← object_allocation_server)
- impure elemental subroutine [f_planarlocalizer_class::planarlocalizer_class_delete](#) (this)
- type(c_planarlocalizer) function [f_planarlocalizer_class::planarlocalizer_class_getcobject](#) (this)
- subroutine [f_planarlocalizer_class::planarlocalizer_class_addplane](#) (this, a_normal, a_distance)
- subroutine [f_planarlocalizer_class::planarlocalizer_class_setnumberofplanes](#) (this, a_number_to_set)
- subroutine [f_planarlocalizer_class::planarlocalizer_class_setplane](#) (this, a_plane_index_to_set, a_← normal, a_distance)
- subroutine [f_planarlocalizer_class::planarlocalizer_class_setfromrectangularcuboid](#) (this, a_lower_pt, a_upper_pt)
- subroutine [f_planarlocalizer_class::planarlocalizer_class_printtoscreen](#) (this)

7.22.1 Detailed Description

This file allows use of the [IRL](#) PlanarLocalizer class through a fortran interface.

7.23 f_planarseparator_class.f90 File Reference

This file allows use of the [IRL](#) PlanarSeparator class through a fortran interface.

Data Types

- type [f_planarseparator_class::c_planarseparator](#)
- type [f_planarseparator_class::planarseparator_type](#)
- interface [f_planarseparator_class::new](#)
- interface [f_planarseparator_class::getcobject](#)
- interface [f_planarseparator_class::addplane](#)
- interface [f_planarseparator_class::setnumberofplanes](#)
- interface [f_planarseparator_class::setplane](#)
- interface [f_planarseparator_class::copy](#)
- interface [f_planarseparator_class::getnumberofplanes](#)
- interface [f_planarseparator_class::getplane](#)
- interface [f_planarseparator_class::isflipped](#)
- interface [f_planarseparator_class::printtoscreen](#)
- interface [f_planarseparator_class::F_PlanarSeparator_new](#)
- interface [f_planarseparator_class::F_PlanarSeparator_newFromObjectAllocationServer](#)
- interface [f_planarseparator_class::F_PlanarSeparator_delete](#)
- interface [f_planarseparator_class::F_PlanarSeparator_addPlane](#)
- interface [f_planarseparator_class::F_PlanarSeparator_setNumberOfPlanes](#)
- interface [f_planarseparator_class::F_PlanarSeparator_setPlane](#)
- interface [f_planarseparator_class::F_PlanarSeparator_copy](#)
- interface [f_planarseparator_class::F_PlanarSeparator_getNumberOfPlanes](#)
- interface [f_planarseparator_class::F_PlanarSeparator_getPlane](#)
- interface [f_planarseparator_class::F_PlanarSeparator_isFlipped](#)
- interface [f_planarseparator_class::F_PlanarSeparator_printToScreen](#)

Modules

- module [f_planarseparator_class](#)

A fortran type class that allows the creation of IRL's PlanarSeparator class along with enabling some of its methods.

Functions/Subroutines

- subroutine [f_planarseparator_class::planarseparator_class_new](#) (this)
- subroutine [f_planarseparator_class::planarseparator_class_newfromobjectallocationserver](#) (this, a←_object_allocation_server)
- impure elemental subroutine [f_planarseparator_class::planarseparator_class_delete](#) (this)
- type(c_planarseparator) function [f_planarseparator_class::planarseparator_class_getcobject](#) (this)
- subroutine [f_planarseparator_class::planarseparator_class_addplane](#) (this, a_normal, a_distance)
- subroutine [f_planarseparator_class::planarseparator_class_setnumberofplanes](#) (this, a_number_to←_set)
- subroutine [f_planarseparator_class::planarseparator_class_setplane](#) (this, a_plane_index_to_set, a←_normal, a_distance)
- subroutine [f_planarseparator_class::planarseparator_class_copy](#) (this, a_other_PlanarSeparator)
- integer(irl_unsignedindex_t) function [f_planarseparator_class::planarseparator_class_getnumberofplanes](#) (this)
- real(irl_double) function, dimension(4) [f_planarseparator_class::planarseparator_class_getplane](#) (this, a_index)
- logical(1) function [f_planarseparator_class::planarseparator_class_isflipped](#) (this)
- subroutine [f_planarsePARATOR_class::planarsePARATOR_class_printtoscreen](#) (this)

7.23.1 Detailed Description

This file allows use of the IRL PlanarSeparator class through a fortran interface.

7.24 f_polygon_class.f90 File Reference

This file contains the Fortran interface for the Polygon class.

Data Types

- type [f_polygon_class::c_polygon](#)
- type [f_polygon_class::polygon_type](#)
- interface [f_polygon_class::new](#)
- interface [f_polygon_class::getcobject](#)
- interface [f_polygon_class::construct](#)
- interface [f_polygon_class::calculatenormal](#)
- interface [f_polygon_class::getlocalizer](#)
- interface [f_polygon_class::reverseptordering](#)
- interface [f_polygon_class::getboundingpts](#)
- interface [f_polygon_class::getnumberofvertices](#)
- interface [f_polygon_class::getpt](#)
- interface [f_polygon_class::getnumberofsimplicesindecomposition](#)
- interface [f_polygon_class::getsimplexfromdecomposition](#)
- interface [f_polygon_class::zeropolygon](#)

- interface `f_polygon_class::calculatenearestptonsurface`
- interface `f_polygon_class::calculatevolume`
- interface `f_polygon_class::calculatesign`
- interface `f_polygon_class::setplaneofexistence`
- interface `f_polygon_class::calculateandsetplaneofexistence`
- interface `f_polygon_class::calculatecentroid`
- interface `f_polygon_class::getplaneofexistence`
- interface `f_polygon_class::printtoscreen`
- interface `f_polygon_class::F_Polygon_new`
- interface `f_polygon_class::F_Polygon_delete`
- interface `f_polygon_class::F_Polygon_construct`
- interface `f_polygon_class::F_Polygon_calculateNormal`
- interface `f_polygon_class::F_Polygon_getLocalizer`
- interface `f_polygon_class::F_Polygon_reversePtOrdering`
- interface `f_polygon_class::F_Polygon_getBoundingPts`
- interface `f_polygon_class::F_Polygon_getNumberOfPts`
- interface `f_polygon_class::F_Polygon_getPt`
- interface `f_polygon_class::F_Polygon_getNumberOfSimplicesInDecomposition`
- interface `f_polygon_class::F_Polygon_getSimplexFromDecomposition`
- interface `f_polygon_class::F_Polygon_zeroPolygon`
- interface `f_polygon_class::F_Polygon_calculateNearestPtOnSurface`
- interface `f_polygon_class::F_Polygon_calculateVolume`
- interface `f_polygon_class::F_Polygon_calculateSign`
- interface `f_polygon_class::F_Polygon_setPlaneOfExistence`
- interface `f_polygon_class::F_Polygon_calculateAndSetPlaneOfExistence`
- interface `f_polygon_class::F_Polygon_getPlaneOfExistence`
- interface `f_polygon_class::F_Polygon_calculateCentroid`
- interface `f_polygon_class::F_Polygon_printToScreen`

Modules

- module `f_polygon_class`

A fortran type class that allows the creation of IRL's Polygon class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_polygon_class::polygon_class_new` (this)
- impure elemental subroutine `f_polygon_class::polygon_class_delete` (this)
- type(c_polygon) function `f_polygon_class::polygon_class_getcobject` (this)
- subroutine `f_polygon_class::polygon_class_construct` (this, a_npts, a_pts)
- real(irl_double) function, dimension(1:3) `f_polygon_class::polygon_class_calculatenormal` (this)
- subroutine `f_polygon_class::polygon_class_getlocalizer` (this, a_planar_localizer)
- subroutine `f_polygon_class::polygon_class_reverseptordering` (this)
- subroutine `f_polygon_class::polygon_class_getboundingpts` (this, a_lower_pt, a_upper_pt)
- integer(irl_unsignedindex_t) function `f_polygon_class::polygon_class_getnumberoftpts` (this)
- real(irl_double) function, dimension(3) `f_polygon_class::polygon_class_getpt` (this, a_index)
- integer(irl_unsignedindex_t) function `f_polygon_class::polygon_class_getnumberofsimplicesindecomposition` (this)
- subroutine `f_polygon_class::polygon_class_getsimplexfromdecomposition` (this, a_tri_number_to_get, a_tri_in_decomposition)
- subroutine `f_polygon_class::polygon_class_zeropolygon` (this)
- real(irl_double) function, dimension(3) `f_polygon_class::polygon_class_calculatenearestptonsurface` (this, a_pt)

- real(irl_double) function **f_polygon_class::polygon_class_calculatevolume** (this)
- real(irl_double) function **f_polygon_class::polygon_class_calculatesign** (this)
- subroutine **f_polygon_class::polygon_class_setplaneofexistence** (this, a_plane)
- subroutine **f_polygon_class::polygon_class_calculateandsetplaneofexistence** (this)
- real(irl_double) function, dimension(4) **f_polygon_class::polygon_class_getplaneofexistence** (this)
- real(irl_double) function, dimension(3) **f_polygon_class::polygon_class_calculatecentroid** (this)
- subroutine **f_polygon_class::polygon_class_printtoscreen** (this)

7.24.1 Detailed Description

This file contains the Fortran interface for the Polygon class.

7.25 f_polyhedron24_class.f90 File Reference

This file contains the Fortran interface for the Polyhedron24 class.

Data Types

- type **f_polyhedron24_class::c_polyhedron24**
- type **f_polyhedron24_class::polyhedron24_type**
- interface **f_polyhedron24_class::new**
- interface **f_polyhedron24_class::getcobject**
- interface **f_polyhedron24_class::construct**
- interface **f_polyhedron24_class::adjustcaptomatchvolume**
- interface **f_polyhedron24_class::getboundingpts**
- interface **f_polyhedron24_class::getpt**
- interface **f_polyhedron24_class::setpt**
- interface **f_polyhedron24_class::F_Polyhedron24_new**
- interface **f_polyhedron24_class::F_Polyhedron24_delete**
- interface **f_polyhedron24_class::F_Polyhedron24_construct**
- interface **f_polyhedron24_class::F_Polyhedron24_adjustCapToMatchVolume**
- interface **f_polyhedron24_class::F_Polyhedron24_getBoundingPts**
- interface **f_polyhedron24_class::F_Polyhedron24_getPt**
- interface **f_polyhedron24_class::F_Polyhedron24_setPt**

Modules

- module **f_polyhedron24_class**

A fortran type class that allows the creation of IRL's Polyhedron24 class along with enabling some of its methods.

Functions/Subroutines

- subroutine **f_polyhedron24_class::polyhedron24_class_new** (this)
- impure elemental subroutine **f_polyhedron24_class::polyhedron24_class_delete** (this)
- type(c_polyhedron24) function **f_polyhedron24_class::polyhedron24_class_getcobject** (this)
- subroutine **f_polyhedron24_class::polyhedron24_class_construct** (this, a_polyhedron24)
- subroutine **f_polyhedron24_class::polyhedron24_class_adjustcaptomatchvolume** (this, a_correct_← signed_volume)
- subroutine **f_polyhedron24_class::polyhedron24_class_getboundingpts** (this, a_lower_pt, a_upper_pt)
- real(irl_double) function, dimension(3) **f_polyhedron24_class::polyhedron24_class_getpt** (this, a_index)
- subroutine **f_polyhedron24_class::polyhedron24_class_setpt** (this, a_index, a_pt)

7.25.1 Detailed Description

This file contains the Fortran interface for the Polyhedron24 class.

7.26 f_polyhedron24_doubles3_class.f90 File Reference

This file contains the Fortran interface for the Polyhedron24_doubles3 class.

Data Types

- type `f_polyhedron24_doubles3_class::c_polyhedron24_doubles3`
- type `f_polyhedron24_doubles3_class::polyhedron24_doubles3_type`
- interface `f_polyhedron24_doubles3_class::new`
- interface `f_polyhedron24_doubles3_class::getcobject`
- interface `f_polyhedron24_doubles3_class::construct`
- interface `f_polyhedron24_doubles3_class::adjustcaptomatchvolume`
- interface `f_polyhedron24_doubles3_class::getboundingpts`
- interface `f_polyhedron24_doubles3_class::getpt`
- interface `f_polyhedron24_doubles3_class::setpt`
- interface `f_polyhedron24_doubles3_class::getdata`
- interface `f_polyhedron24_doubles3_class::setdata`
- interface `f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_new`
- interface `f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_delete`
- interface `f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_construct`
- interface `f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_adjustCapToMatchVolume`
- interface `f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getBoundingPts`
- interface `f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getPt`
- interface `f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setPt`
- interface `f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_getData`
- interface `f_polyhedron24_doubles3_class::F_Polyhedron24_doubles3_setData`

Modules

- module `f_polyhedron24_doubles3_class`

A fortran type class that allows the creation of IRL's Polyhedron24_doubles3 class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_new` (this)
- impure elemental subroutine `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_delete` (this)
- type(`c_polyhedron24_doubles3`) function `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_getcobject` (this)
- subroutine `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_construct` (this, a_\leftarrow polyhedron24, a_data)
- subroutine `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_adjustcaptomatchvolume` (this, a_correct_signed_volume)
- subroutine `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_getboundingpts` (this, a_lower_pt, a_upper_pt)
- real(irl_double) function, dimension(3) `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_getpt` (this, a_index)
- subroutine `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_setpt` (this, a_index, a_pt)
- real(irl_double) function, dimension(3) `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_getdata` (this, a_index)
- subroutine `f_polyhedron24_doubles3_class::polyhedron24_doubles3_class_setdata` (this, a_index, a_data)

7.26.1 Detailed Description

This file contains the Fortran interface for the Polyhedron24_doubles3 class.

7.27 f_r2pneighborhood_rectangularcuboid_class.f90 File Reference

This file contains functions reproducing the functionality of the [IRL](#) class R2PNeighborhood_RectangularCuboid. The purpose of this is to allow building the stencil through references to then supply to obtain a PlanarSeparator using the R2P method.

Data Types

- type [f_r2pneighborhood_rectangularcuboid_class::c_r2pneighborhood_rectangularcuboid](#)
- type [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_type](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::new](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::getcobject](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::setszie](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::setmember](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::addmember](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::emptyneighborhood](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::setcenterofstencil](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::setsurfacearea](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_new](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_delete](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setSize](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setMember](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_addMember](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_emptyNeighborhood](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setCenterOfStencil](#)
- interface [f_r2pneighborhood_rectangularcuboid_class::F_R2PNeighborhood_RectangularCuboid_setSurfaceArea](#)

Modules

- module [f_r2pneighborhood_rectangularcuboid_class](#)
A fortran type class to provide the functionality of R2PNeighborhood_RectangularCuboid.

Functions/Subroutines

- subroutine [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_class_new](#) (this)
- impure elemental subroutine [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_class_delete](#) (this)
- type(c_r2pneighborhood_rectangularcuboid) function [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_class_getcobject](#) (this)
- subroutine [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_class_setszie](#) (this, a_size)
- subroutine [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_class_setmember](#) (this, a_rectangular_cuboid, a_separated_volume_moments, a_index)
- subroutine [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_class_addmember](#) (this, a_rectangular_cuboid, a_separated_volume_moments)
- subroutine [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_class_emptyneighborhood](#) (this)
- subroutine [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_class_setcenterofstencil](#) (this, a_center_cell_index)
- subroutine [f_r2pneighborhood_rectangularcuboid_class::r2pneighborhood_rectangularcuboid_class_setsurfacearea](#) (this, a_surface_area)

7.27.1 Detailed Description

This file contains functions reproducing the functionality of the [IRL](#) class R2PNeighborhood_RectangularCuboid. The purpose of this is to allow building the stencil through references to then supply to obtain a PlanarSeparator using the R2P method.

7.28 f_rectangularcuboid_class.f90 File Reference

This file contains the Fortran interface for the RectangularCuboid class.

Data Types

- type [f_rectangularcuboid_class::c_rectangularcuboid](#)
- type [f_rectangularcuboid_class::rectangularcuboid_type](#)
- interface [f_rectangularcuboid_class::new](#)
- interface [f_rectangularcuboid_class::getobject](#)
- interface [f_rectangularcuboid_class::construct](#)
- interface [f_rectangularcuboid_class::construct_2pt](#)
- interface [f_rectangularcuboid_class::calculatevolume](#)
- interface [f_rectangularcuboid_class::getboundingpts](#)
- interface [f_rectangularcuboid_class::F_RectangularCuboid_new](#)
- interface [f_rectangularcuboid_class::F_RectangularCuboid_delete](#)
- interface [f_rectangularcuboid_class::F_RectangularCuboid_construct](#)
- interface [f_rectangularcuboid_class::F_RectangularCuboid_construct_2pt](#)
- interface [f_rectangularcuboid_class::F_RectangularCuboid_calculateVolume](#)
- interface [f_rectangularcuboid_class::F_RectangularCuboid_getBoundingPts](#)

Modules

- module [f_rectangularcuboid_class](#)

A fortran type class that allows the creation of [IRL](#)'s RectangularCuboid class along with enabling some of its methods.

Functions/Subroutines

- subroutine [f_rectangularcuboid_class::rectangularcuboid_class_new](#) (this)
- impure elemental subroutine [f_rectangularcuboid_class::rectangularcuboid_class_delete](#) (this)
- type(c_rectangularcuboid) function [f_rectangularcuboid_class::rectangularcuboid_class_getobject](#) (this)
- subroutine [f_rectangularcuboid_class::rectangularcuboid_class_construct](#) (this, a_transported_cell)
- subroutine [f_rectangularcuboid_class::rectangularcuboid_class_construct_2pt](#) (this, a_lower_pt, a_upper_pt)
- real(irl_double) function [f_rectangularcuboid_class::rectangularcuboid_class_calculatevolume](#) (this)
- subroutine [f_rectangularcuboid_class::rectangularcuboid_class_getboundingpts](#) (this, a_lower_pt, a_upper_pt)

7.28.1 Detailed Description

This file contains the Fortran interface for the RectangularCuboid class.

7.29 f_sepvm_class.f90 File Reference

This file contains the Fortran interface for volume moments classes.

Data Types

- type `f_sepvm_class::c_sepvm`
- type `f_sepvm_class::sepvm_type`
- interface `f_sepvm_class::new`
- interface `f_sepvm_class::construct`
- interface `f_sepvm_class::getobject`
- interface `f_sepvm_class::normalizebyvolume`
- interface `f_sepvm_class::multiplybyvolume`
- interface `f_sepvm_class::getvolume`
- interface `f_sepvm_class::getcentroid`
- interface `f_sepvm_class::getvolumeptr`
- interface `f_sepvm_class::getcentroidptr`
- interface `f_sepvm_class::F_SepVM_new`
- interface `f_sepvm_class::F_SepVM_delete`
- interface `f_sepvm_class::F_SepVM_construct`
- interface `f_sepvm_class::F_SepVM_normalizeByVolume`
- interface `f_sepvm_class::F_SepVM_multiplyByVolume`
- interface `f_sepvm_class::F_SepVM_getVolume`
- interface `f_sepvm_class::F_SepVM_getCentroid`
- interface `f_sepvm_class::F_SepVM_getVolumePtr`
- interface `f_sepvm_class::F_SepVM_getCentroidPtr`

Modules

- module `f_sepvm_class`

A fortran type class that allows the creation of IRL's SeparatedMoments<VolumeMoments> class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_sepvm_class::sepvm_class_new` (this)
- impure elemental subroutine `f_sepvm_class::sepvm_class_delete` (this)
- type(`c_sepvm`) function `f_sepvm_class::sepvm_class_getobject` (this)
- subroutine `f_sepvm_class::sepvm_class_construct` (this, a_moments_list)
- subroutine `f_sepvm_class::sepvm_class_normalizebyvolume` (this)
- subroutine `f_sepvm_class::sepvm_class_multiplybyvolume` (this)
- real(`irl_double`) function `f_sepvm_class::sepvm_class_getvolume` (this, a_index)
- real(`irl_double`) function, dimension(3) `f_sepvm_class::sepvm_class_getcentroid` (this, a_index)
- real(`irl_double`) function, pointer `f_sepvm_class::sepvm_class_getvolumeptr` (this, a_index)
- real(`irl_double`) function, dimension(:), pointer `f_sepvm_class::sepvm_class_getcentroidptr` (this, a_index)

7.29.1 Detailed Description

This file contains the Fortran interface for volume moments classes.

7.30 f_sepvm_doubles3_class.f90 File Reference

This file contains the Fortran interface for volume moments classes.

Data Types

- type `f_sepvm_doubles3_class::c_sepvm_doubles3`
- type `f_sepvm_doubles3_class::sepvm_doubles3_type`
- interface `f_sepvm_doubles3_class::new`
- interface `f_sepvm_doubles3_class::getcobject`
- interface `f_sepvm_doubles3_class::normalizebyvolume`
- interface `f_sepvm_doubles3_class::multiplybyvolume`
- interface `f_sepvm_doubles3_class::getvolume`
- interface `f_sepvm_doubles3_class::getcentroid`
- interface `f_sepvm_doubles3_class::getdata`
- interface `f_sepvm_doubles3_class::getvolumeptr`
- interface `f_sepvm_doubles3_class::getcentroidptr`
- interface `f_sepvm_doubles3_class::F_SepVM_doubles3_new`
- interface `f_sepvm_doubles3_class::F_SepVM_doubles3_delete`
- interface `f_sepvm_doubles3_class::F_SepVM_doubles3_normalizeByVolume`
- interface `f_sepvm_doubles3_class::F_SepVM_doubles3_multiplyByVolume`
- interface `f_sepvm_doubles3_class::F_SepVM_doubles3_getVolume`
- interface `f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroid`
- interface `f_sepvm_doubles3_class::F_SepVM_doubles3_getData`
- interface `f_sepvm_doubles3_class::F_SepVM_doubles3_getVolumePtr`
- interface `f_sepvm_doubles3_class::F_SepVM_doubles3_getCentroidPtr`

Modules

- module `f_sepvm_doubles3_class`

A fortran type class that allows the creation of IRL's SeparatedMoments< VolumeMoments> class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_sepvm_doubles3_class::sepvm_doubles3_class_new` (this)
- impure elemental subroutine `f_sepvm_doubles3_class::sepvm_doubles3_class_delete` (this)
- type(`c_sepvm_doubles3`) function `f_sepvm_doubles3_class::sepvm_doubles3_class_getcobject` (this)
- subroutine `f_sepvm_doubles3_class::sepvm_doubles3_class_normalizebyvolume` (this)
- subroutine `f_sepvm_doubles3_class::sepvm_doubles3_class_multiplybyvolume` (this)
- real(irl_double) function `f_sepvm_doubles3_class::sepvm_doubles3_class_getvolume` (this, a_index)
- real(irl_double) function, dimension(3) `f_sepvm_doubles3_class::sepvm_doubles3_class_getcentroid` (this, a_index)
- real(irl_double) function, dimension(3) `f_sepvm_doubles3_class::sepvm_doubles3_class_getdata` (this, a_index)
- real(irl_double) function, pointer `f_sepvm_doubles3_class::sepvm_doubles3_class_getvolumeptr` (this, a_index)
- real(irl_double) function, dimension(:,), pointer `f_sepvm_doubles3_class::sepvm_doubles3_class_getcentroidptr` (this, a_index)

7.30.1 Detailed Description

This file contains the Fortran interface for volume moments classes.

7.31 f_serializer.f90 File Reference

This file deals with serializing [IRL](#) class objects into byte buffers. This is usually done before parallel communication via MPI using MPI_BYTE.

Data Types

- interface [f_serializer::serializeandpack](#)
- interface [f_serializer::unpackandstore](#)
- interface [f_serializer::F_Serializer_serializeAndPack_PlanarSeparator_ByteBuffer](#)
- interface [f_serializer::F_Serializer_unpackAndStore_PlanarSeparator_ByteBuffer](#)

Modules

- module [f_serializer](#)

This module contains mappings to the [IRL](#) C interface that deal with serializing [IRL](#) class objects into an array of bytes and packing them into a byte buffer.

Functions/Subroutines

- subroutine [f_serializer::serializeandpack_planarseparator_bytebuffer](#) (a_separator, a_byte_buffer)
- subroutine [f_serializer::unpackandstore_planarseparator_bytebuffer](#) (a_separator, a_byte_buffer)

7.31.1 Detailed Description

This file deals with serializing [IRL](#) class objects into byte buffers. This is usually done before parallel communication via MPI using MPI_BYTE.

7.32 f_tagged_accumlistedvm_vman_class.f90 File Reference

This file contains the Fortran interface for volume moments classes.

Data Types

- type `f_tagged_accumlistedvm_vman_class::c_tagged_accumlistedvm_vman`
- type `f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_type`
- interface `f_tagged_accumlistedvm_vman_class::new`
- interface `f_tagged_accumlistedvm_vman_class::getcobject`
- interface `f_tagged_accumlistedvm_vman_class::getlistatindex`
- interface `f_tagged_accumlistedvm_vman_class::append`
- interface `f_tagged_accumlistedvm_vman_class::clear`
- interface `f_tagged_accumlistedvm_vman_class::getsize`
- interface `f_tagged_accumlistedvm_vman_class::gettagforindex`
- interface `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_new`
- interface `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_delete`
- interface `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getListAtIndex`
- interface `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_append`
- interface `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_clear`
- interface `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getSize`
- interface `f_tagged_accumlistedvm_vman_class::F_Tagged_AccumListedVM_VMAN_getTagForIndex`

Modules

- module `f_tagged_accumlistedvm_vman_class`

A fortran type class that allows the creation of `IRL`'s `TaggedAccumulatedListedVolumeMomentsM<Volume←MomentsAndNormal>` class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_class_new` (this)
- impure elemental subroutine `f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_class_delete` (this)
- type(`c_tagged_accumlistedvm_vman`) function `f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_class_getcobject` (this)
- subroutine `f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_class_getlistatindex` (this, a_index, a_other_list)
- subroutine `f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_class_append` (this, a_other_list)
- subroutine `f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_class_clear` (this)
- integer(`irl_unsignedindex_t`) function `f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_class_getsize` (this)
- integer(`irl_unsignedindex_t`) function `f_tagged_accumlistedvm_vman_class::tagged_accumlistedvm_vman_class_gettagforindex` (this, a_index)

7.32.1 Detailed Description

This file contains the Fortran interface for volume moments classes.

7.33 f_tagged_accumvm_sepvm_class.f90 File Reference

This file contains the Fortran interface for volume moments classes.

Data Types

- type `f_tagged_accumvm_sepvm_class::c_tagged_accumvm_sepvm`
- type `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_type`
- interface `f_tagged_accumvm_sepvm_class::new`
- interface `f_tagged_accumvm_sepvm_class::getcobject`
- interface `f_tagged_accumvm_sepvm_class::normalizebyvolume`
- interface `f_tagged_accumvm_sepvm_class::multiplybyvolume`
- interface `f_tagged_accumvm_sepvm_class::getvolumeatindex`
- interface `f_tagged_accumvm_sepvm_class::getcentroidatindex`
- interface `f_tagged_accumvm_sepvm_class::getvolumeattag`
- interface `f_tagged_accumvm_sepvm_class::getcentroidattag`
- interface `f_tagged_accumvm_sepvm_class::getvolumepratindex`
- interface `f_tagged_accumvm_sepvm_class::getcentroidptratindex`
- interface `f_tagged_accumvm_sepvm_class::getsize`
- interface `f_tagged_accumvm_sepvm_class::gettagforindex`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_new`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_delete`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_normalizeByVolume`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_multiplyByVolume`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtIndex`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtIndex`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumeAtTag`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidAtTag`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getVolumePtrAtIndex`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getCentroidPtrAtIndex`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getSize`
- interface `f_tagged_accumvm_sepvm_class::F_Tagged_AccumVM_SepVM_getTagForIndex`

Modules

- module `f_tagged_accumvm_sepvm_class`

A fortran type class that allows the creation of IRL's AccumulatedVolumeMomentsM<SeparatedMoments<Volume←Moments>> class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_new` (this)
- impure elemental subroutine `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_delete` (this)
- type(`c_tagged_accumvm_sepvm`) function `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_getcobject` (this)
- subroutine `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_normalizebyvolume` (this)
- subroutine `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_multiplybyvolume` (this)
- real(irl_double) function `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_getvolumeatindex` (this, a_list_index, a_index)
- real(irl_double) function, dimension(3) `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_getcentroidatindex` (this, a_list_index, a_index)
- real(irl_double) function `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_getvolumeattag` (this, a_tag, a_index)

- real(irl_double) function, dimension(3) `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_<class_getcentroidattag` (this, a_tag, a_index)
- real(irl_double) function, pointer `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class->getvolumepratindex` (this, a_list_index, a_index)
- real(irl_double) function, dimension(:), pointer `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_getcentroidptratindex` (this, a_list_index, a_index)
- integer(irl_unsignedindex_t) function `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_getsize` (this)
- integer(irl_unsignedindex_t) function `f_tagged_accumvm_sepvm_class::tagged_accumvm_sepvm_class_gettagforindex` (this, a_index)

7.33.1 Detailed Description

This file contains the Fortran interface for volume moments classes.

7.34 f_tagged_accumvm_vm_class.f90 File Reference

This file contains the Fortran interface for volume moments classes.

Data Types

- type `f_tagged_accumvm_vm_class::c_tagged_accumvm_vm`
- type `f_tagged_accumvm_vm_class::tagged_accumvm_vm_type`
- interface `f_tagged_accumvm_vm_class::new`
- interface `f_tagged_accumvm_vm_class::getobject`
- interface `f_tagged_accumvm_vm_class::normalizebyvolume`
- interface `f_tagged_accumvm_vm_class::multiplybyvolume`
- interface `f_tagged_accumvm_vm_class::getvolumeatindex`
- interface `f_tagged_accumvm_vm_class::getcentroidatindex`
- interface `f_tagged_accumvm_vm_class::getvolumepratindex`
- interface `f_tagged_accumvm_vm_class::getcentroidptratindex`
- interface `f_tagged_accumvm_vm_class::getsize`
- interface `f_tagged_accumvm_vm_class::gettagforindex`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_new`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_delete`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_normalizeByVolume`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_multiplyByVolume`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumeAtIndex`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidAtIndex`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getVolumePtrAtIndex`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getCentroidPtrAtIndex`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getSize`
- interface `f_tagged_accumvm_vm_class::F_Tagged_AccumVM_VM_getTagForIndex`

Modules

- module `f_tagged_accumvm_vm_class`

A fortran type class that allows the creation of IRL's AccumulatedVolumeMomentsM<VolumeMoments> class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_new` (this)
- impure elemental subroutine `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_delete` (this)
- type(`c_tagged_accumvm_vm`) function `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_getcobject` (this)
- subroutine `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_normalizebyvolume` (this)
- subroutine `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_multiplybyvolume` (this)
- real(irl_double) function `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_getvolumeatindex` (this, a_list_index)
- real(irl_double) function, dimension(3) `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_getcentroidatindex` (this, a_list_index)
- real(irl_double) function, pointer `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_getvolumepratindex` (this, a_list_index)
- real(irl_double) function, dimension(:, pointer `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_getcentroidptratindex` (this, a_list_index)
- integer(irl_unsignedindex_t) function `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_getsize` (this)
- integer(irl_unsignedindex_t) function `f_tagged_accumvm_vm_class::tagged_accumvm_vm_class_gettagforindex` (this, a_index)

7.34.1 Detailed Description

This file contains the Fortran interface for volume moments classes.

7.35 f_tet_class.f90 File Reference

This file contains the Fortran interface for the Tet class.

Data Types

- type `f_tet_class::c_tet`
- type `f_tet_class::tet_type`
- interface `f_tet_class::new`
- interface `f_tet_class::getcobject`
- interface `f_tet_class::construct`
- interface `f_tet_class::getboundingpts`
- interface `f_tet_class::F_Tet_new`
- interface `f_tet_class::F_Tet_delete`
- interface `f_tet_class::F_Tet_construct`
- interface `f_tet_class::F_Tet_getBoundingPts`

Modules

- module `f_tet_class`

A fortran type class that allows the creation of IRL's Tet class along with enabling some of its methods.

Functions/Subroutines

- subroutine `f_tet_class::tet_class_new` (this)
- impure elemental subroutine `f_tet_class::tet_class_delete` (this)
- type(`c_tet`) function `f_tet_class::tet_class_getcobject` (this)
- subroutine `f_tet_class::tet_class_construct` (this, `a_Tet_pts`)
- subroutine `f_tet_class::tet_class_getboundingpts` (this, `a_lower_pt`, `a_upper_pt`)

7.35.1 Detailed Description

This file contains the Fortran interface for the Tet class.

7.36 f_tri_class.f90 File Reference

This file contains the Fortran interface for the Tri class.

Data Types

- type `f_tri_class::c_tri`
- type `f_tri_class::tri_type`
- interface `f_tri_class::new`
- interface `f_tri_class::getcobject`
- interface `f_tri_class::construct`
- interface `f_tri_class::getvertices`
- interface `f_tri_class::calculatevolume`
- interface `f_tri_class::calculatecentroid`
- interface `f_tri_class::calculatenormal`
- interface `f_tri_class::getlocalizer`
- interface `f_tri_class::reverseptordering`
- interface `f_tri_class::getboundingpts`
- interface `f_tri_class::calculatesign`
- interface `f_tri_class::setplaneofexistence`
- interface `f_tri_class::calculateandsetplaneofexistence`
- interface `f_tri_class::getplaneofexistence`
- interface `f_tri_class::F_Tri_new`
- interface `f_tri_class::F_Tri_delete`
- interface `f_tri_class::F_Tri_construct`
- interface `f_tri_class::F_Tri_getVertices`
- interface `f_tri_class::F_Tri_calculateVolume`
- interface `f_tri_class::F_Tri_calculateCentroid`
- interface `f_tri_class::F_Tri_calculateNormal`
- interface `f_tri_class::F_Tri_getLocalizer`
- interface `f_tri_class::F_Tri_reversePtOrdering`
- interface `f_tri_class::F_Tri_getBoundingPts`
- interface `f_tri_class::F_Tri_calculateSign`
- interface `f_tri_class::F_Tri_setPlaneOfExistence`
- interface `f_tri_class::F_Tri_calculateAndSetPlaneOfExistence`
- interface `f_tri_class::F_Tri_getPlaneOfExistence`

Modules

- module [f_tri_class](#)

A fortran type class that allows the creation of IRL's Tri class along with enabling some of its methods.

Functions/Subroutines

- subroutine [f_tri_class::tri_class_new](#) (this)
- impure elemental subroutine [f_tri_class::tri_class_delete](#) (this)
- type(c_tri) function [f_tri_class::tri_class_getcobject](#) (this)
- subroutine [f_tri_class::tri_class_construct](#) (this, a_pts)
- real(irl_double) function, dimension(1:3, 1:3) [f_tri_class::tri_class_getvertices](#) (this)
- real(irl_double) function [f_tri_class::tri_class_calculatevolume](#) (this)
- real(irl_double) function, dimension(1:3) [f_tri_class::tri_class_calculatecentroid](#) (this)
- real(irl_double) function, dimension(1:3) [f_tri_class::tri_class_calculatenormal](#) (this)
- subroutine [f_tri_class::tri_class_getlocalizer](#) (this, a_planar_localizer)
- subroutine [f_tri_class::tri_class_reverseptordering](#) (this)
- subroutine [f_tri_class::tri_class_getboundingpts](#) (this, a_lower_pt, a_upper_pt)
- real(irl_double) function [f_tri_class::tri_class_calculatesign](#) (this)
- subroutine [f_tri_class::tri_class_setplaneofexistence](#) (this, a_plane)
- subroutine [f_tri_class::tri_class_calculateandsetplaneofexistence](#) (this)
- real(irl_double) function, dimension(4) [f_tri_class::tri_class_getplaneofexistence](#) (this)

7.36.1 Detailed Description

This file contains the Fortran interface for the Tri class.

7.37 f_vman_class.f90 File Reference

This file contains the Fortran interface for volume moments classes.

Data Types

- type [f_vman_class::c_vman](#)
- type [f_vman_class::vman_type](#)
- interface [f_vman_class::new](#)
- interface [f_vman_class::getcobject](#)
- interface [f_vman_class::getvolume](#)
- interface [f_vman_class::getcentroid](#)
- interface [f_vman_class::getnormal](#)
- interface [f_vman_class::normalizebyvolume](#)
- interface [f_vman_class::multiplybyvolume](#)
- interface [f_vman_class::F_VMAN_new](#)
- interface [f_vman_class::F_VMAN_delete](#)
- interface [f_vman_class::F_VMAN_getVolume](#)
- interface [f_vman_class::F_VMAN_getCentroid](#)
- interface [f_vman_class::F_VMAN_getNormal](#)
- interface [f_vman_class::F_VMAN_normalizeByVolume](#)
- interface [f_vman_class::F_VMAN_multiplyByVolume](#)

Modules

- module [f_vman_class](#)

A fortran type class that allows the creation of [IRL](#)'s AccumulatedListedVolumeMomentsM< VolumeMomentsAndNormal> class along with enabling some of its methods.

Functions/Subroutines

- subroutine [f_vman_class::vman_class_new](#) (this)
- impure elemental subroutine [f_vman_class::vman_class_delete](#) (this)
- type(c_vman) function [f_vman_class::vman_class_getcobject](#) (this)
- real(irl_double) function [f_vman_class::vman_class_getvolume](#) (this)
- real(irl_double) function, dimension(3) [f_vman_class::vman_class_getcentroid](#) (this)
- real(irl_double) function, dimension(3) [f_vman_class::vman_class_getnormal](#) (this)
- subroutine [f_vman_class::vman_class_normalizebyvolume](#) (this)
- subroutine [f_vman_class::vman_class_multiplybyvolume](#) (this)

7.37.1 Detailed Description

This file contains the Fortran interface for volume moments classes.

7.38 f_volumefractionmatching.f90 File Reference

This file deals with setting the distances to each plane in a planar reconstruction to match a given volume fraction for the provided cell.

Data Types

- interface [f_volumefractionmatching::setdistancetomatchvolumefraction](#)
- interface [f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS](#)
- interface [f_volumefractionmatching::F_setDistanceToMatchVolumeFraction_RC_PS_DefTol](#)

Modules

- module [f_volumefractionmatching](#)

This module contains mappings to the [IRL](#) C interface that deals with setting the distance to each plane in a reconstruction to recreate the volume fraction on the provided polyhedron.

Functions/Subroutines

- subroutine [f_volumefractionmatching::setdistancetomatchvolumefraction_rc_ps](#) (a_rectangular_cuboid, a_volume_fraction, a_planar_separator, a_volume_fraction_tolerance)
- subroutine [f_volumefractionmatching::setdistancetomatchvolumefraction_rc_ps_deftol](#) (a_rectangular_cuboid, a_volume_fraction, a_planar_separator)

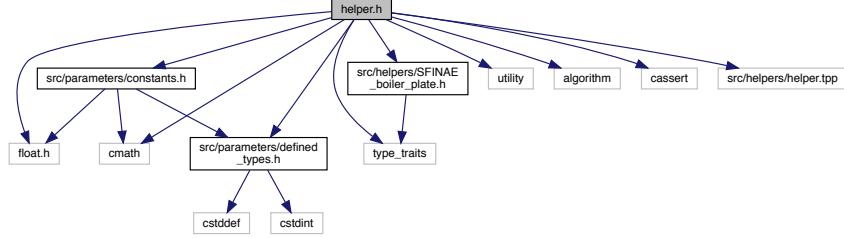
7.38.1 Detailed Description

This file deals with setting the distances to each plane in a planar reconstruction to match a given volume fraction for the provided cell.

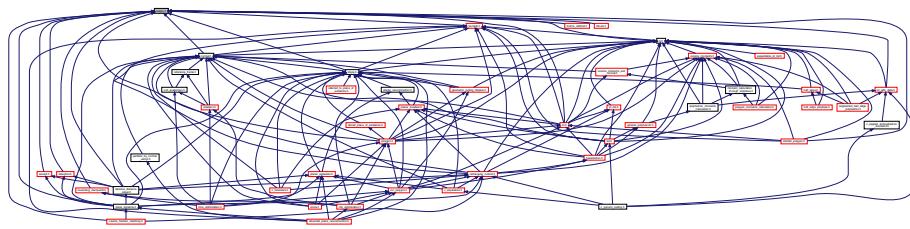
7.39 helper.h File Reference

```
#include <float.h>
#include <type_traits>
#include <utility>
#include <algorithm>
#include <cassert>
#include <cmath>
#include "src/helpers/SFINAE_boiler_plate.h"
#include "src/parameters/constants.h"
#include "src/parameters/defined_types.h"
#include "src/helpers/helper.tpp"
```

Include dependency graph for helper.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [IRL::Expr< E >](#)
- class [IRL::vector_sum< V1, V2 >](#)

Namespaces

- [IRL](#)

Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Functions

- `double IRL::safelyTiny (const double a_value)`
Takes max between abs(a_value) and DBL_MIN while preserving sign.
- `double IRL::safelyEpsilon (const double a_value)`
Takes max between abs(a_value) and DBL_EPSILON while preserving sign.
- `double IRL::safelySmall (const double a_value, const double a_small_value)`
Takes max between abs(a_value) and the_smallest_value while preserving sign.
- `double IRL::clipBetween (const double a_smallest_value, const double a_value, const double a_largest_value)`
Clip value to lay between the min and max values.
- `bool IRL::wantPurelyInternal (const double a_internal_fraction)`
Returns whether the liquid volume fraction indicates full liquid.
- `bool IRL::wantPurelyExternal (const double a_internal_fraction)`
Returns whether the liquid volume fraction indicates full gas.
- `void IRL::sort3Ascending (double *a_items)`
Sort 3 doubles in an array into ascending order via insertion.
- `void IRL::sort3Descending (double *a_items)`
Sort 3 doubles in an array into descending order via insertion.
- `void IRL::sort3AscendingTracked (double *a_items, int *a_original_index)`
Sort 3 doubles in an array into ascending order via insertion. The original location of the item is also tracked to allow reversing.
- `void IRL::sort3DescendingTracked (double *a_items, int *a_original_index)`
Sort 3 doubles in an array into descending order via insertion. The original location of the item is also tracked to allow reversing.
- template<class CarriedType , class DictatingType >
`void IRL::sortAscendingBasedOnOtherArray (CarriedType *a_carried_array, DictatingType *a_dictating_array)`
This function sorts a_carried_array based on the elements in a_dictating_array using insertion sort.
- template<class CarriedType , class DictatingType >
`void IRL::sortDescendingBasedOnOtherArray (CarriedType *a_carried_array, DictatingType *a_dictating_array)`
This function sorts a_carried_array based on the elements in a_dictating_array using insertion sort.
- template<typename V1 , typename V2 >
`enable_if_t< std::is_base_of< Expr< V1 >, V1 >::value && std::is_base_of< Expr< V2 >, V2 >::value, vector_sum< V1, V2 > > IRL::operator+ (const Expr< V1 > &x, const Expr< V2 > &y)`

7.39.1 Detailed Description

Helper functions that perform small but necessary functions that make coding simpler and prevents code bloat.

First, the function declarations are given. Afterwards, the inlined function definitions are given.

7.40 irl_fortran_interface.f90 File Reference

This file serves to provide a single include directive when using the `IRL` fortran interface.

Modules

- module [irl_fortran_interface](#)

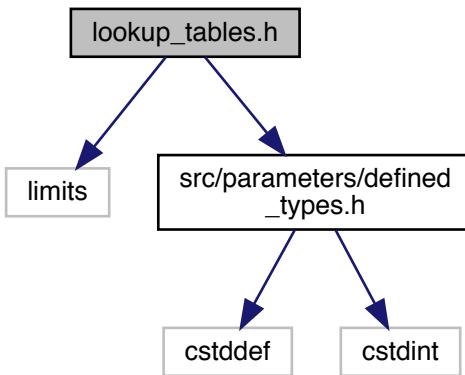
This is just a master wrapper for the entire [IRL](#) fortran interface. For information about each module, view the documentation for the module itself.

7.40.1 Detailed Description

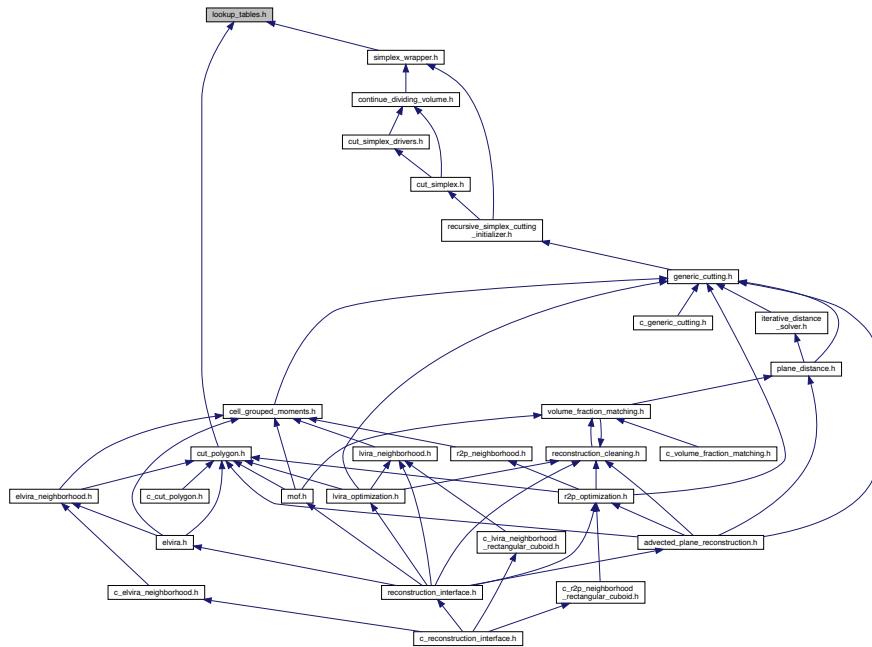
This file serves to provide a single include directive when using the [IRL](#) fortran interface.

7.41 lookup_tables.h File Reference

```
#include <limits>
#include "src/parameters/defined_types.h"
Include dependency graph for lookup_tables.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- **IRL**

Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Variables

- static constexpr LookupIndex_t [IRL::cut_tet_by_plane::number_of_tets_after_cut \[16\]](#)
The total number of resulting single-side tets after cutting a tet by a plane.
- static constexpr LookupIndex_t [IRL::cut_tet_by_plane::number_of_negative_tets_after_cut \[16\]](#)
Number of negative tets ($d <= 0$) formed after cutting a tet by a plane.
- static constexpr LookupIndex_t [IRL::cut_tet_by_plane::number_of_new_vertices_after_cut \[16\]](#)
The number of intersections between the plane and tet edges.
- static constexpr LookupIndex_t [IRL::cut_tet_by_plane::cut_vertices \[16\]\[2\]\[4\]](#)
The vertices used in interpolation along the tet edge to find the point of intersection with a plane.
- static constexpr LookupIndex_t [IRL::cut_tet_by_plane::verts_for_tets \[16\]\[6\]\[4\]](#)
The vertices used to create the new tets in `number_of_tets_after_cut []`
- static constexpr LookupIndex_t [IRL::cut_tri_by_plane::number_of_tris_after_cut \[8\]](#)
The total number of resulting single-side tris after cutting a tri by a plane.
- static constexpr LookupIndex_t [IRL::cut_tri_by_plane::number_of_negative_tris_after_cut \[8\]](#)
Number of negative tris ($d <= 0$) formed after cutting a tet by a plane.
- static constexpr LookupIndex_t [IRL::cut_tri_by_plane::number_of_new_vertices_after_cut \[8\]](#)
The number of intersections between the plane and tri edges.
- static constexpr LookupIndex_t [IRL::cut_tri_by_plane::cut_vertices \[8\]\[2\]\[2\]](#)
The vertices used in interpolation along the tri edge to find the point of intersection with a plane.
- static constexpr LookupIndex_t [IRL::cut_tri_by_plane::verts_for_tris \[8\]\[3\]\[3\]](#)

The vertices used to create the new tris in number_of_tris_after_cut []

- static constexpr LookupIndex_t IRL::cut_plane_by_cuboid::number_of_plane_intersections_with_cuboid [256]

Number of intersections that occur for a plane intersecting a rectangular cuboid, according to a unique case ID.

- static constexpr LookupIndex_t IRL::cut_plane_by_cuboid::cuboid_cut_vertices [256][2][6]

The vertices used in interpolation along the cuboid edge to find the point of intersection with a plane.

- static constexpr LookupIndex_t IRL::cut_cube_by_plane_into_tets::number_of_tets_under_plane [256]

The number of tets underneath the plane when a plane cuts a rectangular cuboid.

- static constexpr LookupIndex_t IRL::cut_cube_by_plane_into_tets::total_number_of_tets [256]

The total number of single-sided tets that will recreate the volume of the rectangular cuboid.

- static constexpr LookupIndex_t IRL::cut_cube_by_plane_into_tets::tet_vertices [256][17][4]

The vertex numbers used in each of the total_number_of_tets for each case.

7.41.1 Detailed Description

This file contains lookup tables that are needed to perform the geoemtric cutting operations. They are used to determine resulting tets from cutting operations and dividing cells into 5 tets. In most cases, the ordering of the numbers in the lookup table below is VERY important and must be kept self-consistent. It is best not to change them unless you are very familiar with the routines and what each lookup table is doing and how it is used.

7.41.2 Variable Documentation

7.41.2.1 cuboid_cut_vertices

```
constexpr LookupIndex_t IRL::cut_plane_by_cuboid::cuboid_cut_vertices[256][2][6] [static]
```

The vertices used in interpolation along the cuboid edge to find the point of intersection with a plane.

This is a list of vertices that are used to interpolate to the intersection point between edges of a cuboid and planes to create the new vertices referred to in number_of_plane_intersections_with_cuboid[].

The three indices from left to right are:

- [cutting_case]
- [vertex_on_the_edge_to_get]
- [cuboid_edge_vertex_that_has_the_intersection]

7.41.2.2 cut_vertices [1/2]

```
constexpr LookupIndex_t IRL::cut_tet_by_plane::cut_vertices[16][2][4] [static]
```

Initial value:

```
= {
    {{NA, NA, NA, NA}, {NA, NA, NA, NA}}, {{0, 0, 0, NA}, {1, 2, 3, NA}},
    {{1, 1, 1, NA}, {0, 2, 3, NA}}, {{0, 0, 1, 1}, {2, 3, 2, 3}},
    {{2, 2, 2, NA}, {0, 1, 3, NA}}, {{0, 0, 2, 2}, {1, 3, 1, 3}},
    {{1, 1, 2, 2}, {0, 3, 0, 3}}, {{0, 1, 2, NA}, {3, 3, 3, NA}},
    {{0, 1, 2, NA}, {3, 3, 3, NA}}, {{1, 1, 2, 2}, {0, 3, 0, 3}},
    {{0, 0, 2, 2}, {1, 3, 1, 3}}, {{2, 2, 2, NA}, {0, 1, 3, NA}},
    {{0, 0, 1, 1}, {2, 3, 2, 3}}, {{1, 1, 1, NA}, {0, 2, 3, NA}},
    {{0, 0, 0, NA}, {1, 2, 3, NA}}, {{NA, NA, NA, NA}, {NA, NA, NA, NA}}}
```

The vertices used in interpolation along the tet edge to find the point of intersection with a plane.

This is a list of vertices that are used to interpolate to the intersection point between tet edges and planes to create the new vertices referred to in `number_of_new_vertices_after_cut []`.

The three indices from left to right are:

- [cutting_case]
- [vertex_on_the_edge_to_get]
- [tet_edge_that_has_an_intersection]

7.41.2.3 cut_vertices [2/2]

```
constexpr LookupIndex_t IRL::cut_tri_by_plane::cut_vertices[8][2][2] [static]
```

Initial value:

```
= {
    {{NA, NA}, {NA, NA}}, {{0, 0}, {1, 2}}, {{1, 1}, {0, 2}},
    {{0, 1}, {2, 2}}, {{0, 1}, {2, 2}}, {{1, 1}, {0, 2}},
    {{0, 0}, {1, 2}}, {{NA, NA}, {NA, NA}}}
```

The vertices used in interpolation along the tri edge to find the point of intersection with a plane.

This is a list of vertices that are used to interpolate to the intersection point between tri edges and planes to create the new vertices referred to in `number_of_new_vertices_after_cut []`.

The three indices from left to right are:

- [cutting_case]
- [vertex_on_the_edge_to_get]
- [tri_edge_that_has_an_intersection]

7.41.2.4 number_of_negative_tets_after_cut

```
constexpr LookupIndex_t IRL::cut_tet_by_plane::number_of_negative_tets_after_cut[16] [static]
```

Initial value:

```
= {  
    1, 3, 3, 3, 3, 3, 3, 1, 3, 3, 1, 3, 1, 1, 0}
```

Number of negative tets ($d \leq 0$) formed after cutting a tet by a plane.

Using this array along with the total number of tets (`number_of_tets_after_cut[]`) the positive tets can also be cycled through. The first index for positive tet is `tet_first_positive = number_of_negative_tets_after_cut[case]`, where the total number of positive tets that exists is `number_of_positive_tets_after_cut[case] = number_of_tets_after_cut[case] - number_of_negative_tets_after_cut[case]`.

The index is `[cutting_case]`, which indicates the sign of each tet vertex.

7.41.2.5 number_of_negative_tris_after_cut

```
constexpr LookupIndex_t IRL::cut_tri_by_plane::number_of_negative_tris_after_cut[8] [static]
```

Initial value:

```
= {  
    1, 2, 2, 1, 2, 1, 1, 0}
```

Number of negative tris ($d \leq 0$) formed after cutting a tet by a plane.

Using this array along with the total number of tris (`number_of_tris_after_cut[]`) the positive tris can also be cycled through. The first index for positive tri is `tri_first_positive = number_of_negative_tris_after_cut[case]`, where the total number of positive tris that exists is `number_of_positive_tris_after_cut[case] = number_of_tris_after_cut[case] - number_of_negative_tris_after_cut[case]`.

The index is `[cutting_case]`, which indicates the sign of each tet vertex.

7.41.2.6 number_of_new_vertices_after_cut [1/2]

```
constexpr LookupIndex_t IRL::cut_tet_by_plane::number_of_new_vertices_after_cut[16] [static]
```

Initial value:

```
= {  
    0, 3, 3, 4, 3, 4, 4, 3, 3, 4, 4, 3, 4, 3, 3, 0}
```

The number of intersections between the plane and tet edges.

This is the number of intersections between the cutting plane and tet edges. When a plane intersects a tet edge, a new vertex will be placed at this location. The new vertices are then used when constructing new tets that lay completely on one side of the cutting plane.

The index is `[cutting_case]`, which indicates the sign of each tet vertex.

7.41.2.7 number_of_new_vertices_after_cut [2/2]

```
constexpr LookupIndex_t IRL::cut_tri_by_plane::number_of_new_vertices_after_cut[8] [static]
```

Initial value:

```
= {  
    0, 2, 2, 2, 2, 2, 2, 0}
```

The number of intersections between the plane and tri edges.

This is the number of intersections between the cutting plane and tet edges. When a plane intersects a tri edge, a new vertex will be placed at this location. The new vertices are then used when constructing new tris that lay completely on one side of the cutting plane.

The index is [cutting_case], which indicates the sign of each tri vertex.

7.41.2.8 number_of_plane_intersections_with_cuboid

```
constexpr LookupIndex_t IRL::cut_plane_by_cuboid::number_of_plane_intersections_with_cuboid[256]  
[static]
```

Initial value:

```
= {  
    0, 3, 3, 4, 3, NA, 4, 5, 3, 4, NA, 5, 4, 5, 5, 4, 3, 4, NA,  
    5, NA, NA, NA, NA, NA, 5, NA, 6, NA, NA, NA, 5, 3, NA, 4, 5, NA, NA,  
    5, 6, NA, NA, NA, NA, NA, NA, 5, 4, 5, 5, 4, NA, NA, NA, 5, NA,  
    NA, NA, 5, NA, NA, NA, NA, 4, 3, NA, NA, NA, 4, NA, 5, NA, NA, NA, NA,  
    5, NA, 6, 5, NA,  
    NA, 4, NA, 5, NA, 5, NA, 4, 5, NA, NA, NA, NA, NA, NA, NA, 5, 4, 5, NA,  
    6, 5, NA, NA, 5, 4, NA, NA, NA, NA, NA, 3, 3, NA, NA, NA, NA, NA,  
    NA, NA, NA, 4, 5, NA, NA, 5, 6, NA, 5, 4, 5, NA, NA, NA, NA, NA, NA,  
    5, 4, NA, 5, NA, 5, NA, 4, NA,  
    NA, NA, NA, NA, NA, 5, 6, NA, 5, NA, NA, NA, NA, NA, 5, NA, 4, NA, NA,  
    NA, 3, 4, NA, NA, NA, 5, NA, NA, 5, NA, NA, 4, 5, 5, 4, 5, NA, NA, NA,  
    NA, NA, NA, NA, NA, NA, 6, 5, NA, NA, 5, 4, NA, 3, 5, NA, NA, NA, NA,  
    6, NA, 5, NA, NA, NA, NA, 5, NA, 4, 3, 4, 5, 5, 4, 5, NA, 4,  
    3, 5, 4, NA, 3, 4, 3, 3, 0}
```

Number of intersections that occur for a plane intersecting a rectangular cuboid, according to a unique case ID.

Point ordering on the rectangular cuboid is important here. Point ordering is as follows. (x,y,z) of vertex

Vertex 0 : (+, -, -) Vertex 1 : (+, +, -) Vertex 2 : (+, +, +) Vertex 3 : (+, -, +) Vertex 4 : (-, -, -) Vertex 5 : (-, +, -) Vertex 6 : (-, +, +) Vertex 7 : (-, -, +)

The index is [cutting_case], which indicates the sign of each tet vertex.

7.41.2.9 number_of_tets_after_cut

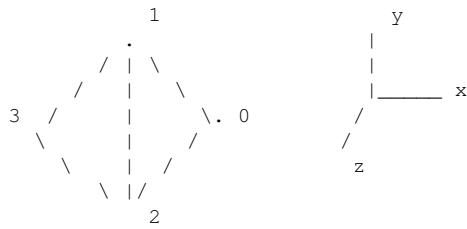
```
constexpr LookupIndex_t IRL::cut_tet_by_plane::number_of_tets_after_cut[16] [static]
```

Initial value:

```
= {  
    1, 4, 4, 6, 4, 6, 6, 4, 4, 6, 6, 4, 6, 4, 4, 1}
```

The total number of resulting single-side tets after cutting a tet by a plane.

Tet vertex ordering, showing positive volume tet



This is the number of single-side tets that result from cutting a tet by a plane. By single-side tet, we mean one that lays entirely below or above the cutting plane. By doing this, the original tet is represented by a collection of tets that lay entirely on a side of the plane. When this plane represents a liquid-gas interface for instance, the new tets are now fully in a single phase, allowing separate liquid and gas quantities to be found for the originally phase-mixed tet.

The index is [cutting_case], which indicates the sign of each tet vertex.

7.41.2.10 number_of_tets_under_plane

```
constexpr LookupIndex_t IRL::cut_cube_by_plane_into_tets::number_of_tets_under_plane[256]
[static]
```

Initial value:

```
= {
  6, 10, 10, 9, 10, NA, 9, 10, 10, 9, NA, 10, 9, 10, 10, 6, 10, 9, NA,
  10, NA, NA, NA, NA, 10, NA, 7, NA, NA, NA, 7, 10, NA, 9, 10, NA, NA,
  10, 7, NA, NA, NA, NA, NA, NA, 7, 9, 10, 10, 6, NA, NA, NA, 7, NA,
  NA, NA, 7, NA, NA, NA, 3, 10, NA, NA, NA, 9, NA, 10, NA, NA, NA, NA, NA,
  10, NA, 7, NA, NA,
  NA, 9, NA, 10, NA, 10, NA, 6, 7, NA, NA, NA, NA, NA, NA, 7, 3, 10, NA,
  7, 7, NA, NA, 7, 3, NA, NA, NA, NA, NA, NA, NA, 1, 10, NA, NA, NA, NA,
  NA, NA, NA, 9, 10, NA, NA, 10, 7, NA, 7, 9, 10, NA, NA, NA, NA, NA, NA,
  10, 6, NA, 7, NA, 7, NA, 3, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
  NA, NA, NA, NA, NA, 10, 7, NA, 7, NA, NA, NA, NA, 7, NA, 3, NA, NA,
  NA, 1, 9, NA, NA, NA, 10, NA, NA, NA, 10, NA, NA, NA, 6, 7, 7, 3, 10,
  NA, NA, NA, NA, NA, NA, NA, 7, 7, NA, NA, 7, 3, NA, 1, 10, NA, NA, NA,
  7, NA, 7, NA, NA, NA, NA, NA, 7, NA, 3, 1, 6, 7, 7, 3, 7, NA, 3,
  1, 7, 3, NA, 1, 3, 1, 1, 0}
```

The number of tets underneath the plane when a plane cuts a rectangular cuboid.

This is a lookup table giving the number of tets needed to represent the volume underneath a plane cutting a rectangular cuboid. This is used with `tet_vertices` to construct the individual tets. A value of NA indicates an impossible case ID when cutting by a plane.

The index is [cutting_case], which indicates the sign of each tet vertex.

7.41.2.11 number_of_tris_after_cut

```
constexpr LookupIndex_t IRL::cut_tri_by_plane::number_of_tris_after_cut[8] [static]
```

Initial value:

```
= {1, 3, 3, 3,
   3, 3, 3, 1}
```

The total number of resulting single-side tris after cutting a tri by a plane.

This is the number of single-side tris that result from cutting a tri by a plane. By single-side tri, we mean one that lays entirely below or above the cutting plane. By doing this, the original tri is represented by a collection of tets that lay entirely on a side of the plane. When this plane represents a liquid-gas interface for instance, the new tri are now fully in a single phase, allowing separate liquid and gas quantities to be found for the originally phase-mixed tri.

The index is [cutting_case], which indicates the sign of each tri vertex.

7.41.2.12 tet_vertices

```
constexpr LookupIndex_t IRL::cut_cube_by_plane_into_tets::tet_vertices[256][17][4] [static]
```

The vertex numbers used in each of the total_number_of_tets for each case.

This is a lookup table that provides the vertex numbers needed to create each of the total_number_of_tets for a individual case. Values of NA indicate elements that should never be accessed.

Tets below the plane are given in `tet_vertices[256][0:number_of_tets_under_planeNA][1:4]. Tets above the plane are given in `tet_vertices[256][number_of_tets_under_plane:total_number_of_tetsNA][1:4].

The two indices from left to right are:

- [cutting_case]
- [number_of_tet_to_get]
- [vertex_in_tet]

7.41.2.13 total_number_of_tets

```
constexpr LookupIndex_t IRL::cut_cube_by_plane_into_tets::total_number_of_tets[256] [static]
```

Initial value:

```
= {
  6, 11, 11, 12, 11, NA, 12, 17, 11, 12, NA, 17, 12, 17, 17, 12, 11, 12, NA,
  17, NA, NA, NA, NA, 17, NA, 14, NA, NA, NA, 17, 11, NA, 12, 17, NA, NA,
  17, 14, NA, NA, NA, NA, NA, NA, 17, 12, 17, 17, 12, NA, NA, NA, 17, NA,
  NA, NA, 17, NA, NA, NA, 12, 11, NA, NA, NA, 12, NA, 17, NA, NA, NA, NA, NA,
  17, NA, 14, 17, NA, NA,
  NA, 12, NA, 17, NA, 17, NA, 12, 17, NA, NA, NA, NA, NA, NA, 17, 12, 17, NA,
  14, 17, NA, NA, 17, 12, NA, NA, NA, NA, NA, NA, 11, 11, NA, NA, NA, NA,
  NA, NA, 12, 17, NA, NA, 17, 14, NA, 17, 12, 17, NA, NA, NA, NA, NA, NA,
  17, 12, NA, 17, NA, 17, NA, 12, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
  NA, NA, NA, NA, 17, 14, NA, 17, NA, NA, NA, NA, NA, 17, NA, 12, NA, NA,
  NA, 11, 12, NA, NA, 17, NA, NA, NA, 17, NA, NA, NA, 12, 17, 17, 12, 17,
  NA, NA, NA, NA, NA, NA, 14, 17, NA, NA, 17, 12, NA, 11, 17, NA, NA, NA,
  14, NA, 17, NA, NA, NA, NA, 17, NA, 12, 11, 12, 17, 17, 12, 17, NA, 12,
  11, 17, 12, NA, 11, 12, 11, 11, 6}
```

The total number of single-sided tets that will recreate the volume of the rectangular cuboid.

This is a lookup table giving the total number of single-sided tets needed to represent the volume of a rectangular cuboid after cut by a plane. The number of positive tets is positive_tets = total_number_of_tets - number_of_tets_under_plane. A value of NA indicates an impossible case ID when cutting by a plane.

The index is [cutting_case], which indicates the sign of each Hexahedron vertex.

7.41.2.14 verts_for_tets

```
constexpr LookupIndex_t IRL::cut_tet_by_plane::verts_for_tets[16][6][4] [static]
```

The vertices used to create the new tets in number_of_tets_after_cut []

This is the vertices of the tet used to create new, single-side tets after cutting by a plane. Here, there are 8 possible vertex numbers (0-7), due to the 4 original tet vertices and the possibility of 4 additional vertices being created during intersection with the plane.

The three indices from left to right are:

- [cutting_case]
- [tet_number_to_get]
- [vertices_that_make_the_tet]

7.41.2.15 verts_for_tris

```
constexpr LookupIndex_t IRL::cut_tri_by_plane::verts_for_tris[8][3][3] [static]
```

Initial value:

```
= {
    {{0, 1, 2}, {NA, NA, NA}, {NA, NA, NA}},
    {{3, 1, 2}, {3, 2, 4}, {3, 4, 0}},
    {{3, 4, 2}, {3, 2, 0}, {3, 1, 4}},
    {{3, 4, 2}, {4, 3, 0}, {4, 0, 1}},
    {{4, 3, 0}, {4, 0, 1}, {3, 4, 2}},
    {{3, 1, 4}, {3, 4, 2}, {3, 2, 0}},
    {{3, 4, 0}, {3, 1, 2}, {3, 2, 4}},
    {{0, 1, 2}, {NA, NA, NA}, {NA, NA, NA}}}
```

The vertices used to create the new tris in `number_of_tris_after_cut[]`

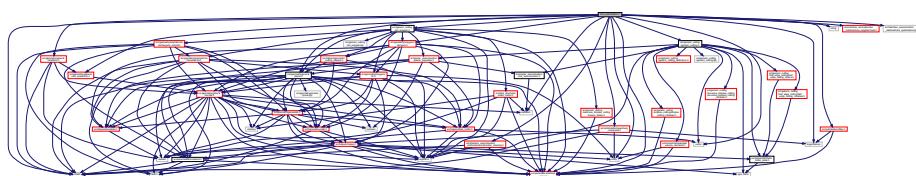
This is the vertices of the tri used to create new, single-side tris after cutting by a plane. Here, there are 5 possible vertex numbers (0-4), due to the 3 original tri vertices and the possibility of 2 additional vertices being created during intersection with the plane.

The three indices from left to right are:

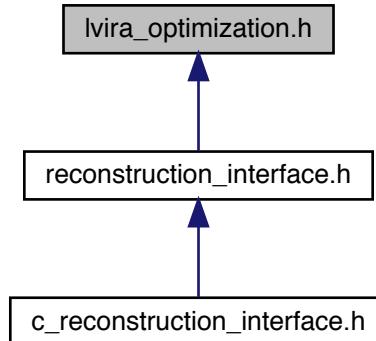
- [cutting_case]
- [tri_number_to_get]
- [vertices_that_make_the_tri]

7.42 lvira_optimization.h File Reference

```
#include <cmath>
#include <iostream>
#include <string>
#include <utility>
#include <vector>
#include <Eigen/Dense>
#include "src/generic_cutting/cut_polygon.h"
#include "src/generic_cutting/generic_cutting.h"
#include "src/geometry/general/plane.h"
#include "src/geometry/general/rotations.h"
#include "src/geometry/general/unit_quaternion.h"
#include "src/geometry/polygons/polygon.h"
#include "src/interface_reconstruction_methods/lvira_neighborhood.h"
#include "src/interface_reconstruction_methods/reconstruction_cleaning.h"
#include "src/optimization/bfgs.h"
#include "src/optimization/levenberg_marquardt.h"
#include "src/parameters/defined_types.h"
#include "src/interface_reconstruction_methods/lvira_optimization.tpp"
Include dependency graph for lvira_optimization.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [IRL::LVIRA_2D< CellType >](#)
LVIRA class for reconstructions in 2D (x-y plane).
- class [IRL::LVIRA_3D< CellType >](#)
LVIRA class for reconstructions in 3 dimensions.
- class [IRL::LVIRACommon< CellType, kColumns >](#)
Class to contain data and methods that will be used in all of the specific LVIRA optimization classes.
- class [IRL::LVIRA_2D< CellType >](#)
LVIRA class for reconstructions in 2D (x-y plane).
- class [IRL::LVIRA_3D< CellType >](#)
LVIRA class for reconstructions in 3 dimensions.
- class [IRL::LVIRADebug< LVIRAType >](#)
This class just calls the LVIRAType functions but allows debug statements to be printed. The solution path is also saved to be exported and visualized.

Namespaces

- [IRL](#)
Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Variables

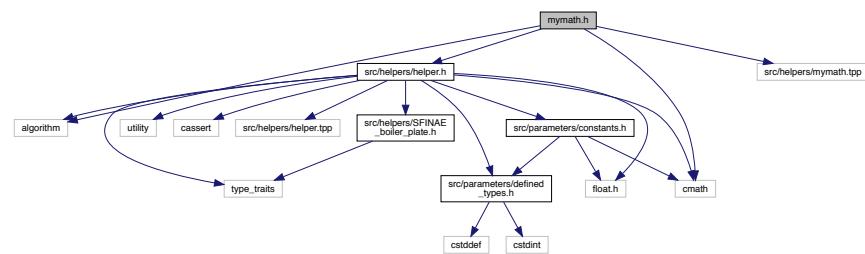
- static constexpr UnsignedIndex_t [IRL::LVIRA_2D_columns](#) = 1
Number of columns for 2D LVIRA optimization, which is equal to parameters being fit.
- static constexpr UnsignedIndex_t [IRL::LVIRA_3D_columns](#) = 2
Number of columns for 3D LVIRA optimization, which is equal to parameters being fit.

7.42.1 Detailed Description

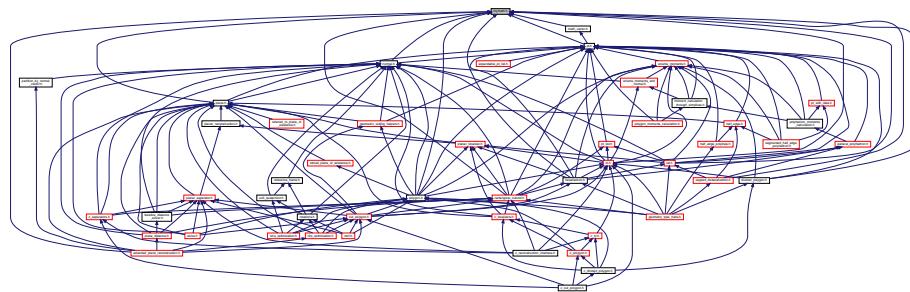
This file contains all LVIRA functions needed to be used during calls to the templated Levenberg Marquardt solver in [optimizers.h](#).

7.43 mymath.h File Reference

```
#include <algorithm>
#include <cmath>
#include "src/helpers/helper.h"
#include "src/helpers/mymath.tpp"
Include dependency graph for mymath.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- [IRL](#)

Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Functions

- `constexpr double IRL::deg2Rad (const double a_degree)`
Convert degrees to radians.
- `constexpr double IRL::rad2Deg (const double a_radian)`
Convert radians to degrees.

- double `IRL::angleNormalize` (const double a_radian)
Normalize a given angle (in radians) to be between 0 and 2 pi.
- double `IRL::signedAngleNormalize` (const double a_radian)
Normalize a given angle (in radians) to be between -2 pi and 2 pi.
- template<class `DataType`>
`double IRL::magnitude` (const `DataType` &a_vector_0, const `DataType` &a_vector_1)
Calculate the magnitude between two 3 element vectors.
- template<class `DataType`>
`double IRL::squaredMagnitude` (const `DataType` &a_vector)
Calculate the squared magnitude between for a 3 element vector.
- template<class `DataType`>
`DataType IRL::crossProduct` (const `DataType` &a_vector_0, const `DataType` &a_vector_1)
Take cross product of two 3-element vectors.
- template<class `DataType`>
`DataType IRL::crossProductNormalized` (const `DataType` &a_vector_0, const `DataType` &a_vector_1)
Cross product of two 3-element vectors then normalized.
- template<class `DataType`>
`double IRL::dotProduct` (const `DataType` &a_vector_0, const `DataType` &a_vector_1)
Dot product between two 3 element vectors.
- template<class `DataType`>
`double IRL::scalarTripleProduct` (const `DataType` &a_vector_0, const `DataType` &a_vector_1, const `DataType` &a_vector_2)
Scalar triple product of 3, 3 element vectors.

7.43.1 Detailed Description

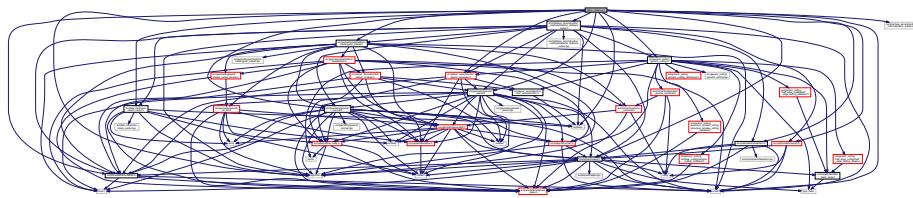
This file contains functions that deal with math like operations like conversions of angle types, dot products, cross products, etc.

First, the function declarations are given. Afterwards, the inlined function definitions are given. Then, templated functions are given.

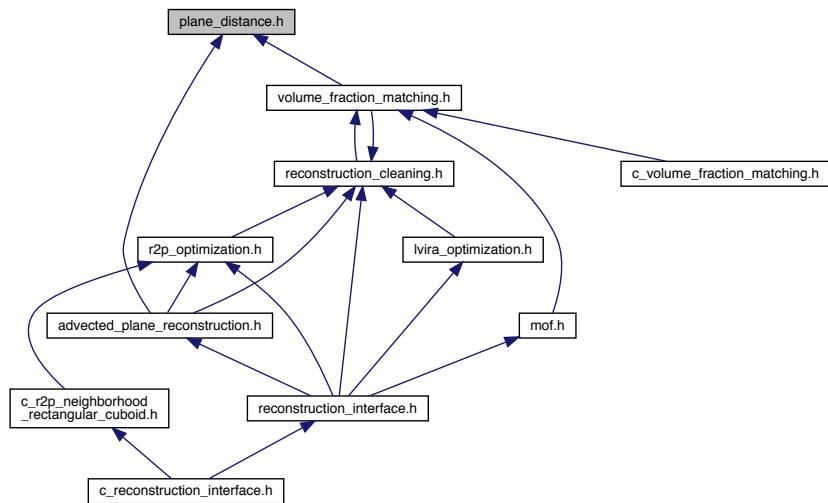
7.44 plane_distance.h File Reference

```
#include <algorithm>
#include <cmath>
#include <iostream>
#include "src/data_structures/stack_vector.h"
#include "src/generic_cutting/generic_cutting.h"
#include "src/geometry/general/normal.h"
#include "src/geometry/general/plane.h"
#include "src/geometry/polyhedrons/rectangular_cuboid.h"
#include "src/helpers/helper.h"
#include "src/interface_reconstruction_methods/iterative_distance_solver.h"
#include "src/optimization/bisection.h"
#include "src/optimization/secant.h"
#include "src/parameters/constants.h"
#include "src/planar_reconstruction/planar_separator.h"
```

```
#include "src/interface_reconstruction_methods/plane_distance.tpp"
Include dependency graph for plane_distance.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- [IRL](#)

Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Functions

- double [IRL::findDistanceOnePlane](#) (const RectangularCuboid &a_rectangular_cuboid, const double a_<
volume_fraction, const Normal &a_normal)

Volume conserving distance-finding routine for single-plane reconstructions.

- double [IRL::getAlpha](#) (const double *a_mm, const double a_VOFo)

Get distance in local coordinate system according to Scardovelli & Zaleski, JCP 164,228-247 (2000)

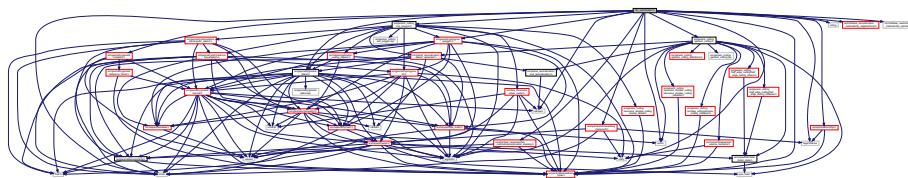
- template<class CellType , class ReconstructionType >
void [IRL::runIterativeSolverForDistance](#) (const CellType &a_cell, const double a_volume_fraction,
ReconstructionType *a_reconstruction, const double a_volume_fraction_tolerance=global_constants::TWO_PLANE_DISTANCE_VOLUME_FRACTION_TOLERANCE)

7.44.1 Detailed Description

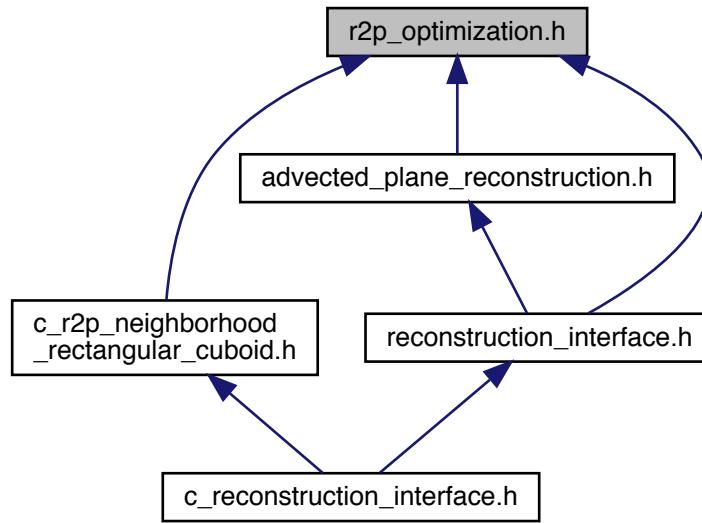
This file contains functions to calculate and set the distance to the plane(s) in provided reconstructions so that a given liquid volume fraction is recreated on the given cell.

7.45 r2p_optimization.h File Reference

```
#include <cmath>
#include <iostream>
#include <string>
#include <utility>
#include <vector>
#include <Eigen/Dense>
#include "src/generic_cutting/cut_polygon.h"
#include "src/generic_cutting/generic_cutting.h"
#include "src/geometry/general/plane.h"
#include "src/geometry/general/rotations.h"
#include "src/geometry/polygons/polygon.h"
#include "src/interface_reconstruction_methods/r2p_neighborhood.h"
#include "src/interface_reconstruction_methods/reconstruction_cleaning.h"
#include "src/optimization/bfgs.h"
#include "src/optimization/levenberg_marquardt.h"
#include "src/parameters/defined_types.h"
#include "src/interface_reconstruction_methods/r2p_optimization.tpp"
Include dependency graph for r2p_optimization.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [IRL::R2P_2D1P< CellType >](#)
R2P class for reconstructions in 2 dimensions with 1 plane (hence 2D1P)
- class [IRL::R2P_3D1P< CellType >](#)
R2P class for reconstructions in 3 dimensions with 1 plane (hence 3D1P)
- class [IRL::R2P_2D2P< CellType >](#)
R2P class for reconstructions in 2 dimensions with 2 plane (hence 2D2P)
- class [IRL::R2P_3D2P< CellType >](#)
R2P class for reconstructions in 3 dimensions with 2 plane (hence 3D2P)
- class [IRL::R2PCommon< CellType, kColumns >](#)
Class to contain data and methods that will be used in all of the specific R2Poptimization.
- class [IRL::R2P_2D1P< CellType >](#)
R2P class for reconstructions in 2 dimensions with 1 plane (hence 2D1P)
- class [IRL::R2P_3D1P< CellType >](#)
R2P class for reconstructions in 3 dimensions with 1 plane (hence 3D1P)
- class [IRL::R2P_2D2P< CellType >](#)
R2P class for reconstructions in 2 dimensions with 2 plane (hence 2D2P)
- class [IRL::R2P_3D2P< CellType >](#)
R2P class for reconstructions in 3 dimensions with 2 plane (hence 3D2P)
- class [IRL::R2PDebug< R2PType >](#)
This class just calls the R2PType functions but allows debug statements to be printed.

Namespaces

- [IRL](#)
Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Variables

- static constexpr UnsignedIndex_t [IRL::R2P_2D1P_ncells](#) = 9
Number of cells for 2D - 1 plane R2P optimization.
- static constexpr UnsignedIndex_t [IRL::R2P_2D1P_rows](#) = 7 * [R2P_2D1P_ncells](#) + 1
Number of rows for 2D - 1 plane R2P optimization, with 7 Entries per cell (Liquid volume, liquid centroid, gas centroid) and the surface area for the cell being reconstructed.
- static constexpr UnsignedIndex_t [IRL::R2P_2D1P_columns](#) = 1
Number of columns for 2D - 1 plane R2P optimization, which is equal to parameters being fit.
- static constexpr UnsignedIndex_t [IRL::R2P_3D1P_ncells](#) = 27
Number of cells for 3D - 1 plane R2P optimization.
- static constexpr UnsignedIndex_t [IRL::R2P_3D1P_rows](#) = 7 * [R2P_3D1P_ncells](#) + 1
Number of rows for 3D - 1 plane R2P optimization, with 7 Entries per cell (Liquid volume, liquid centroid, gas centroid) and the surface area for the cell being reconstructed.
- static constexpr UnsignedIndex_t [IRL::R2P_3D1P_columns](#) = 2
Number of columns for 3D - 1 plane R2P optimization, which is equal to parameters being fit.
- static constexpr UnsignedIndex_t [IRL::R2P_2D2P_ncells](#) = 9
Number of cells for 2D - 2 plane R2P optimization.
- static constexpr UnsignedIndex_t [IRL::R2P_2D2P_rows](#) = 7 * [R2P_2D2P_ncells](#) + 1
Number of rows for 2D - 2 plane R2P optimization, with 7 Entries per cell (Liquid volume, liquid centroid, gas centroid) and the surface area for the cell being reconstructed.
- static constexpr UnsignedIndex_t [IRL::R2P_2D2P_columns](#) = 4
Number of columns for 2D - 2 plane R2P optimization, which is equal to parameters being fit.
- static constexpr UnsignedIndex_t [IRL::R2P_3D2P_ncells](#) = 27
Number of cells for 3D - 2 plane R2P optimization.
- static constexpr UnsignedIndex_t [IRL::R2P_3D2P_rows](#) = 7 * [R2P_3D2P_ncells](#) + 1
Number of rows for 3D - 2 plane R2P optimization, with 7 Entries per cell (Liquid volume, liquid centroid, gas centroid) and the surface area for the cell being reconstructed.
- static constexpr UnsignedIndex_t [IRL::R2P_3D2P_columns](#) = 6
Number of columns for 3D - 2 plane R2P optimization, which is equal to parameters being fit.

7.45.1 Detailed Description

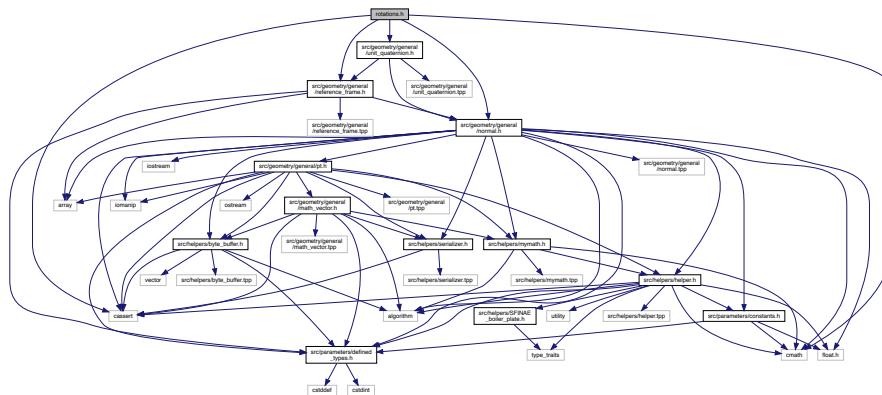
This file contains all R2P functions needed to be used during calls to the templated Levenberg Marquardt solver in [optimizers.h](#).

First, the function declarations are given. Afterwards, the inlined function definitions are given. Then, templated functions are given.

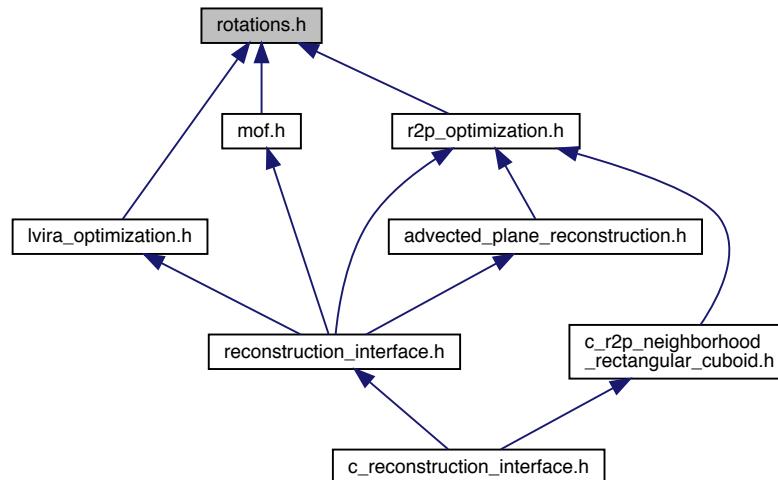
7.46 rotations.h File Reference

```
#include <cassert>
#include <cmath>
#include "src/geometry/general/normal.h"
#include "src/geometry/general/reference_frame.h"
```

```
#include "src/geometry/general/unit_quaternion.h"
Include dependency graph for rotations.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- [IRL](#)

Decomposing of a half-edge data structure representing an initial object into one separated by a series of planar reconstructions.

Functions

- UnitQuaternion [IRL::rotateNormalOntoNormal](#) (const Normal &a_normal_original, const Normal &a_normal_destination, double *a_rotation_amount, Normal *a_rotation_axis)

Return the quaternion that would rotate a_normal_original onto a_normal_destination.

- UnitQuaternion [IRL::rotateNormalOntoNormal](#) (const Normal &a_normal_original, const Normal &a_normal_destination)
Same as other function, however does not return the angle of rotation or rotation axis.
- ReferenceFrame [IRL::getOrthonormalSystem](#) (const Normal &a_normal)
Create an orthonormal reference frame with axis 3 being a_normal.
- Normal [IRL::getSharedNormal](#) (const Normal &a_normal_0, const Normal &a_normal_1, double *a_half_rotation_angle, Normal *a_rotation_axis)
Get normal that is halfway between a_normal_0 and a_normal_1. Return through pointer the rotation to the shared normal and the axis of rotation.
- Normal [IRL::getSharedNormal](#) (const Normal &a_normal_0, const Normal &a_normal_1)
Same as other function, however does not return the half-angle of rotation or rotation axis.

7.46.1 Detailed Description

This file contains classes and functions that deal with getting reference frames and performing projections and rotations into new spaces.

First, class definitions will appear. Below that, inlined function definitions will be given.

Index

__attribute__
IRL, 106

ARBITRARILY_LARGE_DISTANCE
constants.h, 633

addCellMomentsToGeometryVector
IRL::R2PCommon, 530

c_BytBuffer, 138

C_CUTTING_METHOD
IRL, 117

c_CappedDodecahedron, 138

c_CappedDodecahedron_doubles3, 139

c_Constants_setMinimumSurfaceAreaToTrack
c_constants.h, 624

c_Constants_setMinimumVolumeToTrack
c_constants.h, 624

c_Constants_setVolumeFractionBounds
c_constants.h, 624

c_Constants_setVolumeFractionToleranceForIterative←
DistanceFinding
c_constants.h, 625

c_DividedPolygon, 140

c_Dodecahedron, 140

c_ELVIRANeighborhood, 141

c_LVIRANeighborhood_RectangularCuboid, 145

c_ListedVM_VMAN, 142

c_LocalizedSeparatorLink, 143

c_LocalizerLink, 144

c_ObjectAllocationServer_LocalizedSeparatorLink, 146

c_ObjectAllocationServer_LocalizerLink, 147

c_ObjectAllocationServer_PlanarLocalizer, 148

c_ObjectAllocationServer_PlanarSeparator, 149

c_PlanarLocalizer, 150

c_PlanarSeparator, 151

c_Polygon, 152

c_Polyhedron24, 153

c_Polyhedron24_doubles3, 153

c_R2PNeighborhood_RectangularCuboid, 154

c_RectangularCuboid, 155

c_SepVM_doubles3, 156

c_SepVM, 155

c_Tagged_AccumListedVM_VMAN, 157

c_Tagged_AccumVM_SepVM, 158

c_Tagged_AccumVM_VM, 159

c_Tet, 159

c_Tri, 160

c_VMAN, 161

c_constants.h, 623

c_Constants_setMinimumSurfaceAreaToTrack,
624

c_Constants_setMinimumVolumeToTrack, 624

c_Constants_setVolumeFractionBounds, 624

c_Constants_setVolumeFractionToleranceFor←
IterativeDistanceFinding, 625

c_cut_polygon.h, 625

c_getPlanePolygonFromReconstruction_Rectangular←
Cuboid_DividedPolygon, 626

c_getPlanePolygonFromReconstruction_Rectangular←
Cuboid_Polygon, 627

c_getReconstructionSurfaceArea_Rectangular←
Cuboid, 627

c_generic_cutting.h, 627

c_getVolumeMoments_setMethod, 630

c_getPlanePolygonFromReconstruction_Rectangular←
Cuboid_DividedPolygon

c_cut_polygon.h, 626

c_getPlanePolygonFromReconstruction_Rectangular←
Cuboid_Polygon

c_cut_polygon.h, 627

c_getReconstructionSurfaceArea_RectangularCuboid
c_cut_polygon.h, 627

c_getVolumeMoments_setMethod

c_generic_cutting.h, 630

c_localizers.h, 630

c_serializer.h, 631

checkPureObject
IRL, 106

constants.h, 632

ARBITRARILY_LARGE_DISTANCE, 633

MAX_PLANES_IN_OBJECT, 633

crossProduct
IRL, 106

crossProductNormalized
IRL, 107

cuboid_cut_vertices
lookup_tables.h, 669

cut_vertices
lookup_tables.h, 669, 670

cutPlaneByHexahedron
IRL, 107

cutPolygonByPlane
IRL, 108

dotProduct
IRL, 108

f_bytbuffer_class, 55

f_bytbuffer_class.f90, 634

f_bytebuffer_class::F_ByteBuffer_dataPtr, 208
 f_bytebuffer_class::F_ByteBuffer_delete, 208
 f_bytebuffer_class::F_ByteBuffer_getSize, 208
 f_bytebuffer_class::F_ByteBuffer_new, 209
 f_bytebuffer_class::F_ByteBuffer_resetBufferPointer,
 209
 f_bytebuffer_class::F_ByteBuffer_setSize, 209
 f_bytebuffer_class::bytebuffer_type, 137
 f_bytebuffer_class::c_bytebuffer, 137
 f_bytebuffer_class::dataptr, 184
 f_bytebuffer_class::getcobject, 293
 f_bytebuffer_class::getsize, 304
 f_bytebuffer_class::new, 469
 f_bytebuffer_class::resetbufferpointer, 544
 f_bytebuffer_class::setsiz, 576
 f_cappeddodecahedron_class, 56
 f_cappeddodecahedron_class.f90, 635
 f_cappeddodecahedron_class::F_CappedDodecahedron
 _adjustCapToMatchVolume, 209
 f_cappeddodecahedron_class::F_CappedDodecahedron
 _construct, 210
 f_cappeddodecahedron_class::F_CappedDodecahedron
 _delete, 210
 f_cappeddodecahedron_class::F_CappedDodecahedron
 _getBoundingPts, 212
 f_cappeddodecahedron_class::F_CappedDodecahedron
 _getPt, 213
 f_cappeddodecahedron_class::F_CappedDodecahedron
 _new, 213
 f_cappeddodecahedron_class::adjustcaptomatchvolume,
 127
 f_cappeddodecahedron_class::c_cappeddodecahedron,
 139
 f_cappeddodecahedron_class::cappeddodecahedron
 _type, 166
 f_cappeddodecahedron_class::construct, 178
 f_cappeddodecahedron_class::getboundingpts, 283
 f_cappeddodecahedron_class::getcobject, 295
 f_cappeddodecahedron_class::getpt, 301
 f_cappeddodecahedron_class::new, 472
 f_cappeddodecahedron_doubles3_class, 56
 f_cappeddodecahedron_doubles3_class.f90, 636
 f_cappeddodecahedron_doubles3_class::F_Capped
 Dodecahedron_doubles3_adjustCapTo
 MatchVolume, 210
 f_cappeddodecahedron_doubles3_class::F_Capped
 Dodecahedron_doubles3_construct, 210
 f_cappeddodecahedron_doubles3_class::F_Capped
 Dodecahedron_doubles3_delete, 211
 f_cappeddodecahedron_doubles3_class::F_Capped
 Dodecahedron_doubles3_getBoundingPts,
 211
 f_cappeddodecahedron_doubles3_class::F_Capped
 Dodecahedron_doubles3_getData, 211
 f_cappeddodecahedron_doubles3_class::F_Capped
 Dodecahedron_doubles3_getPt, 211
 f_cappeddodecahedron_doubles3_class::F_Capped
 Dodecahedron_doubles3_new, 212
 f_cappeddodecahedron_doubles3_class::F_Capped
 Dodecahedron_doubles3_setData, 212
 f_cappeddodecahedron_doubles3_class::F_Capped
 Dodecahedron_doubles3_setPt, 212
 f_cappeddodecahedron_doubles3_class::adjustcaptomatchvolume,
 126
 f_cappeddodecahedron_doubles3_class::c_cappeddodecahedron
 _doubles3, 139
 f_cappeddodecahedron_doubles3_class::cappeddodecahedron
 _doubles3_type, 165
 f_cappeddodecahedron_doubles3_class::construct, 179
 f_cappeddodecahedron_doubles3_class::getboundingpts,
 284
 f_cappeddodecahedron_doubles3_class::getcobject,
 291
 f_cappeddodecahedron_doubles3_class::getdata, 295
 f_cappeddodecahedron_doubles3_class::getpt, 301
 f_cappeddodecahedron_doubles3_class::new, 467
 f_cappeddodecahedron_doubles3_class::setpt, 575
 f_constants, 57
 f_constants.f90, 637
 f_constants::F_Constants_setMinimumSurfaceArea
 ToTrack, 213
 f_constants::F_Constants_setMinimumVolumeToTrack,
 213
 f_constants::F_Constants_setVolumeFractionBounds,
 214
 f_constants::F_Constants_setVolumeFractionTolerance
 ForDistanceFinding, 214
 f_cutpolygon, 58
 f_cutpolygon.f90, 637
 f_cutpolygon::F_getPlanePolygonFromReconstruction
 _RC_DivPoly, 221
 f_cutpolygon::F_getPlanePolygonFromReconstruction
 _RC_Poly, 221
 f_cutpolygon::F_getReconstructionSurfaceArea_RC,
 222
 f_cutpolygon::getplanepolygonfromreconstruction, 301
 f_cutpolygon::getreconstructionsurfacearea, 303
 f_definedtypes, 59
 f_dividedpolygon_class, 59
 f_dividedpolygon_class.f90, 638
 f_dividedpolygon_class::F_DividedPolygon_calculate
 AndSetPlaneOfExistence, 214
 f_dividedpolygon_class::F_DividedPolygon_calculate
 Normal, 214
 f_dividedpolygon_class::F_DividedPolygon_calculate
 Sign, 215
 f_dividedpolygon_class::F_DividedPolygon_calculate
 SurfaceArea, 215
 f_dividedpolygon_class::F_DividedPolygon_construct,
 215
 f_dividedpolygon_class::F_DividedPolygon_construct
 FromPolygon, 215
 f_dividedpolygon_class::F_DividedPolygon_delete, 216
 f_dividedpolygon_class::F_DividedPolygon_get
 BoundingPts, 216

f_dividedpolygon_class::F_DividedPolygon_get←
 Localizer, 216
f_dividedpolygon_class::F_DividedPolygon_get←
 NumberOfPts, 216
f_dividedpolygon_class::F_DividedPolygon_get←
 NumberOfSimplicesInDecomposition, 217
f_dividedpolygon_class::F_DividedPolygon_getPlane←
 OfExistence, 217
f_dividedpolygon_class::F_DividedPolygon_getPt, 217
f_dividedpolygon_class::F_DividedPolygon_get←
 SimplexFromDecomposition, 217
f_dividedpolygon_class::F_DividedPolygon_new, 218
f_dividedpolygon_class::F_DividedPolygon_printTo←
 Screen, 218
f_dividedpolygon_class::F_DividedPolygon_reset←
 Centroid, 218
f_dividedpolygon_class::F_DividedPolygon_reversePt←
 Ordering, 218
f_dividedpolygon_class::F_DividedPolygon_setPlane←
 OfExistence, 219
f_dividedpolygon_class::F_DividedPolygon_zero←
 Polygon, 219
f_dividedpolygon_class::c_dividedpolygon, 140
f_dividedpolygon_class::calculateandsetplaneofexistence,
 162
f_dividedpolygon_class::calculatenormal, 163
f_dividedpolygon_class::calculatesign, 164
f_dividedpolygon_class::calculatesurfacearea, 164
f_dividedpolygon_class::construct, 180
f_dividedpolygon_class::constructfrompolygon, 181
f_dividedpolygon_class::dividedpolygon_type, 189
f_dividedpolygon_class::getboundingpts, 285
f_dividedpolygon_class::getcobject, 293
f_dividedpolygon_class::getlocalizer, 297
f_dividedpolygon_class::getnumberofsimplicesindecomposition,
 299
f_dividedpolygon_class::getnumberofvertices, 300
f_dividedpolygon_class::getplaneofexistence, 301
f_dividedpolygon_class::getpt, 302
f_dividedpolygon_class::getsimplexfromdecomposition,
 303
f_dividedpolygon_class::new, 468
f_dividedpolygon_class::printtoscreen, 504
f_dividedpolygon_class::resetcentroid, 545
f_dividedpolygon_class::reverseptordering, 547
f_dividedpolygon_class::setplaneofexistence, 574
f_dividedpolygon_class::zeropolygon, 622
f_dodecahedron_class, 61
f_dodecahedron_class.f90, 640
f_dodecahedron_class::F_Dodecahedron_construct,
 219
f_dodecahedron_class::F_Dodecahedron_delete, 219
f_dodecahedron_class::F_Dodecahedron_getBounding←
 Pts, 220
f_dodecahedron_class::F_Dodecahedron_new, 220
f_dodecahedron_class::c_dodecahedron, 140
f_dodecahedron_class::construct, 179
f_dodecahedron_class::dodecahedron_type, 192
f_dodecahedron_class::getboundingpts, 284
f_dodecahedron_class::getcobject, 291
f_dodecahedron_class::new, 466
f_elviraneighborhood_class, 61
f_elviraneighborhood_class::F_ELVIRANeighborhood←
 _delete, 220
f_elviraneighborhood_class::F_ELVIRANeighborhood←
 _new, 220
f_elviraneighborhood_class::F_ELVIRANeighborhood←
 _setMember, 221
f_elviraneighborhood_class::F_ELVIRANeighborhood←
 _setSize, 221
f_elviraneighborhood_class::c_elviraneighborhood, 142
f_elviraneighborhood_class::elviraneighborhood_type,
 203
f_elviraneighborhood_class::getcobject, 291
f_elviraneighborhood_class::new, 465
f_elviraneighborhood_class::setmember, 572
f_elviraneighborhood_class::setsize, 575
f_geometriccuttinghelpers, 62
f_geometriccuttinghelpers.f90, 640
f_geometriccuttinghelpers::F_isPtInternal_PL, 227
f_geometriccuttinghelpers::F_isPtInternal_PS, 227
f_geometriccuttinghelpers::isptinternal, 412
f_getvolumemoments, 62
f_getvolumemoments.f90, 641
f_getvolumemoments::F_GNVM_CD_By_LSL_For_S←
 VM, 222
f_getvolumemoments::F_GNVM_CD_By_LSL_For←
 TagAccumVM_SVM, 222
f_getvolumemoments::F_GNVM_CDWD3_By_LSL←
 For_SVMAD3, 222
f_getvolumemoments::F_GNVM_D_By_LSL_For_SVM,
 223
f_getvolumemoments::F_GNVM_D_By_LSL_For←
 TagAccumVM_SVM, 223
f_getvolumemoments::F_GNVM_D_By_PS_For_SVM,
 223
f_getvolumemoments::F_GNVM_P24_By_LSL_For←
 SVM, 223
f_getvolumemoments::F_GNVM_P24WD3_By_LSL←
 For_SVMAD3, 224
f_getvolumemoments::F_GNVM_Poly_By_PL_For_V,
 224
f_getvolumemoments::F_GNVM_RC_By_PS_For_S←
 VM, 224
f_getvolumemoments::F_GNVM_RC_By_PS_For_V,
 224
f_getvolumemoments::F_GNVM_Tet_By_LSL_For_S←
 VM, 225
f_getvolumemoments::F_GNVM_Tri_By_LL_For_Tag←
 AVM_VM, 225
f_getvolumemoments::F_GNVM_Tri_By_PL_For_V,
 225
f_getvolumemoments::F_GVM_CD_By_LSL_For_SVM,
 225
f_getvolumemoments::F_GVM_D_By_LSL_For_SVM,
 226

f_getvolumemoments::F_GVM_P24_By_LSL_For_SVM, 226
 f_getvolumemoments::F_GVM_Tri_By_LL_For_TagALVM_VMAN, 226
 f_getvolumemoments::F_GVM_setMethod, 226
 f_getvolumemoments::getnormalizedvolumemoments, 298
 f_getvolumemoments::getvolumemoments, 308
 f_getvolumemoments::getvolumemoments_setmethod, 311
 f_listedvmm_vman_class, 64
 f_listedvmm_vman_class::F_ListedVM_VMAN_append, 227
 f_listedvmm_vman_class::F_ListedVM_VMAN_clear, 227
 f_listedvmm_vman_class::F_ListedVM_VMAN_delete, 228
 f_listedvmm_vman_class::F_ListedVM_VMAN_erase, 228
 f_listedvmm_vman_class::F_ListedVM_VMAN_get Moments, 228
 f_listedvmm_vman_class::F_ListedVM_VMAN_getSize, 228
 f_listedvmm_vman_class::F_ListedVM_VMAN_new, 229
 f_listedvmm_vman_class::F_ListedVM_VMAN_zero NormalComponent, 229
 f_listedvmm_vman_class::append, 129
 f_listedvmm_vman_class::c_listedvmm_vman, 143
 f_listedvmm_vman_class::clear, 174
 f_listedvmm_vman_class::erase, 204
 f_listedvmm_vman_class::getcobject, 292
 f_listedvmm_vman_class::getmoments, 298
 f_listedvmm_vman_class::getsize, 304
 f_listedvmm_vman_class::listedvmm_vman_type, 430
 f_listedvmm_vman_class::new, 469
 f_listedvmm_vman_class::zeronormalcomponent, 622
 f_localizedseparatorlink_class, 65
 f_localizedseparatorlink_class.f90, 642
 f_localizedseparatorlink_class::F_LocalizedSeparator Link_delete, 229
 f_localizedseparatorlink_class::F_LocalizedSeparator Link_getId, 229
 f_localizedseparatorlink_class::F_LocalizedSeparator Link_new, 230
 f_localizedseparatorlink_class::F_LocalizedSeparator Link_newFromObjectAllocationServer, 230
 f_localizedseparatorlink_class::F_LocalizedSeparator Link_setEdgeConnectivity, 230
 f_localizedseparatorlink_class::F_LocalizedSeparator Link_setEdgeConnectivityNull, 230
 f_localizedseparatorlink_class::F_LocalizedSeparator Link_setId, 231
 f_localizedseparatorlink_class::c_localizedseparatorlink, 143
 f_localizedseparatorlink_class::getcobject, 293
 f_localizedseparatorlink_class::getid, 296
 f_localizedseparatorlink_class::localizedseparatorlink _type, 434
 f_localizedseparatorlink_class::new, 470
 f_localizedseparatorlink_class::setedgeconnectivity, 570
 f_localizedseparatorlink_class::setedgeconnectivitynull, 571
 f_localizedseparatorlink_class::setid, 572
 f_localizerlink_class, 65
 f_localizerlink_class.f90, 643
 f_localizerlink_class::F_LocalizerLink_delete, 231
 f_localizerlink_class::F_LocalizerLink_getId, 231
 f_localizerlink_class::F_LocalizerLink_new, 231
 f_localizerlink_class::F_LocalizerLink_newFrom ObjectAllocationServer, 232
 f_localizerlink_class::F_LocalizerLink_setEdge Connectivity, 232
 f_localizerlink_class::F_LocalizerLink_setEdge ConnectivityNull, 232
 f_localizerlink_class::F_LocalizerLink_setId, 232
 f_localizerlink_class::c_localizerlink, 144
 f_localizerlink_class::getcobject, 294
 f_localizerlink_class::getid, 296
 f_localizerlink_class::localizerlink_type, 435
 f_localizerlink_class::new, 470
 f_localizerlink_class::setedgeconnectivity, 571
 f_localizerlink_class::setedgeconnectivitynull, 571
 f_localizerlink_class::setid, 572
 f_lviraneighborhood_rectangularcuboid_class, 66
 f_lviraneighborhood_rectangularcuboid_class::F_LVI RANeighborhood_RectangularCuboid_add Member, 233
 f_lviraneighborhood_rectangularcuboid_class::F_LVI VIRANeighborhood_RectangularCuboid_delete, 233
 f_lviraneighborhood_rectangularcuboid_class::F_LVI VIRANeighborhood_RectangularCuboid_emptyNeighborhood, 233
 f_lviraneighborhood_rectangularcuboid_class::F_LVI IRANeighborhood_RectangularCuboid_new, 233
 f_lviraneighborhood_rectangularcuboid_class::F_LVI RANeighborhood_RectangularCuboid_set CenterOfStencil, 234
 f_lviraneighborhood_rectangularcuboid_class::F_LVI RANeighborhood_RectangularCuboid_set Member, 234
 f_lviraneighborhood_rectangularcuboid_class::F_LVI RANeighborhood_RectangularCuboid_set Size, 234
 f_lviraneighborhood_rectangularcuboid_class::addmember, 125
 f_lviraneighborhood_rectangularcuboid_class::c_lviraneighborhood_rectangularcuboid, 145
 f_lviraneighborhood_rectangularcuboid_class::emptyneighborhood, 204
 f_lviraneighborhood_rectangularcuboid_class::getcobject, 295
 f_lviraneighborhood_rectangularcuboid_class::lviraneighborhood_rectangularcuboid_type, 450
 f_lviraneighborhood_rectangularcuboid_class::new, 471

f_lviraneighborhood_rectangularcuboid_class::setcenterofstencil, 569
f_lviraneighborhood_rectangularcuboid_class::setmember, f_objectallocationserver_planarseparator_class::F_delete, 236
f_lviraneighborhood_rectangularcuboid_class::setsize, 573
f_lviraneighborhood_rectangularcuboid_class::setsize, 576
f_objectallocationserver_localizedseparatorlink_class, 67
f_objectallocationserver_localizedseparatorlink_class.f90, 644
f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparatorLink_delete, 234
f_objectallocationserver_localizedseparatorlink_class::F_ObjectAllocationServer_LocalizedSeparatorLink_new, 235
f_objectallocationserver_localizedseparatorlink_class::c_objectallocationserver_localizedseparatorlink, 146
f_objectallocationserver_localizedseparatorlink_class::getobject, 288
f_objectallocationserver_localizedseparatorlink_class::new, 465
f_objectallocationserver_localizedseparatorlink_class::objectallocationserver_localizedseparatorlink_type, 476
f_objectallocationserver_localizerlink_class, 68
f_objectallocationserver_localizerlink_class.f90, 645
f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_delete, 235
f_objectallocationserver_localizerlink_class::F_ObjectAllocationServer_LocalizerLink_new, 235
f_objectallocationserver_localizerlink_class::c_objectallocationserver_localizerlink, 146
f_objectallocationserver_localizerlink_class::getobject, 289
f_objectallocationserver_localizerlink_class::new, 470
f_objectallocationserver_localizerlink_class::objectallocationserver_localizerlink_type, 477
f_objectallocationserver_planarlocalizer_class, 68
f_objectallocationserver_planarlocalizer_class.f90, 646
f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_delete, 235
f_objectallocationserver_planarlocalizer_class::F_ObjectAllocationServer_PlanarLocalizer_new, 236
f_objectallocationserver_planarlocalizer_class::c_objectallocationserver_planarlocalizer, 147
f_objectallocationserver_planarlocalizer_class::getobject, 289
f_objectallocationserver_planarlocalizer_class::new, 465
f_objectallocationserver_planarlocalizer_class::objectallocationserver_planarlocalizer_type, 477
f_objectallocationserver_planarseparator_class, 69
f_objectallocationserver_planarseparator_class.f90, 647
f_objectallocationserver_planarseparator_class::F_delete, 236
f_objectallocationserver_planarseparator_class::F_getobject, 289
f_objectallocationserver_planarseparator_class::F_new, 469
f_objectallocationserver_planarseparator_class::F_setFromRectangularCuboid, 238
f_objectallocationserver_planarseparator_class::F_SetNumberOfPlanes, 238
f_objectallocationserver_planarseparator_class::F_SetPlane, 238
f_planarlocalizer_class, 69
f_planarlocalizer_class.f90, 647
f_planarlocalizer_class::F_PlanarLocalizer_addPlane, 236
f_planarlocalizer_class::F_PlanarLocalizer_delete, 237
f_planarlocalizer_class::F_PlanarLocalizer_new, 237
f_planarlocalizer_class::F_PlanarLocalizer_newFromObjectAllocationServer, 237
f_planarlocalizer_class::F_PlanarLocalizer_printToScreen, 237
f_planarlocalizer_class::F_PlanarLocalizer_setFromRectangularCuboid, 238
f_planarlocalizer_class::F_PlanarLocalizer_setNumberOfPlanes, 238
f_planarlocalizer_class::F_PlanarLocalizer_setPlane, 238
f_planarlocalizer_class::addplane, 126
f_planarlocalizer_class::c_planarlocalizer, 150
f_planarlocalizer_class::getobject, 290
f_planarlocalizer_class::new, 465
f_planarlocalizer_class::planarlocalizer_type, 484
f_planarlocalizer_class::printtoscreen, 504
f_planarlocalizer_class::setfromrectangularcuboid, 571
f_planarlocalizer_class::setnumberofplanes, 573
f_planarlocalizer_class::setplane, 573
f_planarseparator_class, 70
f_planarseparator_class.f90, 648
f_planarseparator_class::F_PlanarSeparator_addPlane, 238
f_planarseparator_class::F_PlanarSeparator_copy, 239
f_planarseparator_class::F_PlanarSeparator_delete, 239
f_planarseparator_class::F_PlanarSeparator_getNumberOfPlanes, 239
f_planarseparator_class::F_PlanarSeparator_getPlane, 239
f_planarseparator_class::F_PlanarSeparator_isFlipped, 240
f_planarseparator_class::F_PlanarSeparator_new, 240
f_planarseparator_class::F_PlanarSeparator_newFromObjectAllocationServer, 240
f_planarseparator_class::F_PlanarSeparator_printToScreen, 240
f_planarseparator_class::F_PlanarSeparator_set

NumberOfPlanes, 241
 f_planarseparator_class::F_PlanarSeparator_setPlane,
 241
 f_planarseparator_class::addplane, 126
 f_planarseparator_class::c_planarseparator, 151
 f_planarseparator_class::copy, 182
 f_planarseparator_class::getcobject, 290
 f_planarseparator_class::getnumberofplanes, 299
 f_planarseparator_class::getplane, 300
 f_planarseparator_class::isflipped, 407
 f_planarseparator_class::new, 466
 f_planarseparator_class::planarseparator_type, 489
 f_planarseparator_class::printtoscreen, 504
 f_planarseparator_class::setnumberofplanes, 573
 f_planarseparator_class::setplane, 574
 f_polygon_class, 71
 f_polygon_class.f90, 649
 f_polygon_class::F_Polygon_calculateAndSetPlane←
 OfExistence, 241
 f_polygon_class::F_Polygon_calculateCentroid, 241
 f_polygon_class::F_Polygon_calculateNearestPtOn←
 Surface, 242
 f_polygon_class::F_Polygon_calculateNormal, 242
 f_polygon_class::F_Polygon_calculateSign, 242
 f_polygon_class::F_Polygon_calculateVolume, 242
 f_polygon_class::F_Polygon_construct, 243
 f_polygon_class::F_Polygon_delete, 243
 f_polygon_class::F_Polygon_getBoundingPts, 243
 f_polygon_class::F_Polygon_getLocalizer, 243
 f_polygon_class::F_Polygon_getNumberOfPts, 244
 f_polygon_class::F_Polygon_getNumberOfSimplices←
 InDecomposition, 244
 f_polygon_class::F_Polygon_getPlaneOfExistence, 244
 f_polygon_class::F_Polygon_getPt, 244
 f_polygon_class::F_Polygon_getSimplexFromDecomposition,
 245
 f_polygon_class::F_Polygon_new, 245
 f_polygon_class::F_Polygon_printToScreen, 245
 f_polygon_class::F_Polygon_reversePtOrdering, 245
 f_polygon_class::F_Polygon_setPlaneOfExistence, 246
 f_polygon_class::F_Polygon_zeroPolygon, 246
 f_polygon_class::c_polygon, 152
 f_polygon_class::calculateandsetplaneofexistence, 162
 f_polygon_class::calculatecentroid, 162
 f_polygon_class::calculatenarestptonsurface, 163
 f_polygon_class::calculatenormal, 163
 f_polygon_class::calculatesign, 164
 f_polygon_class::calculatevolume, 165
 f_polygon_class::construct, 179
 f_polygon_class::getboundingpts, 284
 f_polygon_class::getcobject, 292
 f_polygon_class::getlocalizer, 297
 f_polygon_class::getnumberofsimplicesindecomposition,
 299
 f_polygon_class::getnumberofvertices, 299
 f_polygon_class::getplaneofexistence, 300
 f_polygon_class::getpt, 302
 f_polygon_class::getsimplexfromdecomposition, 303
 f_polygon_class::new, 466
 f_polygon_class::polygon_type, 492
 f_polygon_class::printtoscreen, 505
 f_polygon_class::reverseptordering, 547
 f_polygon_class::setplaneofexistence, 574
 f_polygon_class::zeropolygon, 622
 f_polyhedron24_class, 72
 f_polyhedron24_class.f90, 651
 f_polyhedron24_class::F_Polyhedron24_adjustCapTo←
 MatchVolume, 246
 f_polyhedron24_class::F_Polyhedron24_construct, 246
 f_polyhedron24_class::F_Polyhedron24_delete, 247
 f_polyhedron24_class::F_Polyhedron24_getBounding←
 Pts, 249
 f_polyhedron24_class::F_Polyhedron24_getPt, 249
 f_polyhedron24_class::F_Polyhedron24_new, 250
 f_polyhedron24_class::F_Polyhedron24_setPt, 250
 f_polyhedron24_class::adjustcaptomatchvolume, 127
 f_polyhedron24_class::c_polyhedron24, 152
 f_polyhedron24_class::construct, 179
 f_polyhedron24_class::getboundingpts, 285
 f_polyhedron24_class::getcobject, 292
 f_polyhedron24_class::getpt, 302
 f_polyhedron24_class::new, 468
 f_polyhedron24_class::polyhedron24_type, 500
 f_polyhedron24_class::setpt, 575
 f_polyhedron24_doubles3_class, 73
 f_polyhedron24_doubles3_class.f90, 652
 f_polyhedron24_doubles3_class::F_Polyhedron24 ←
 doubles3_adjustCapToMatchVolume, 247
 f_polyhedron24_doubles3_class::F_Polyhedron24 ←
 doubles3_construct, 247
 f_polyhedron24_doubles3_class::F_Polyhedron24 ←
 doubles3_delete, 247
 f_polyhedron24_doubles3_class::F_Polyhedron24 ←
 doubles3_getBoundingPts, 248
 f_polyhedron24_doubles3_class::F_Polyhedron24 ←
 doubles3_getData, 248
 f_polyhedron24_doubles3_class::F_Polyhedron24 ←
 doubles3_getPt, 248
 f_polyhedron24_doubles3_class::F_Polyhedron24 ←
 doubles3_new, 248
 f_polyhedron24_doubles3_class::F_Polyhedron24 ←
 doubles3_setData, 249
 f_polyhedron24_doubles3_class::F_Polyhedron24 ←
 doubles3_setPt, 249
 f_polyhedron24_doubles3_class::adjustcaptomatchvolume,
 127
 f_polyhedron24_doubles3_class::c_polyhedron24 ←
 doubles3, 153
 f_polyhedron24_doubles3_class::construct, 180
 f_polyhedron24_doubles3_class::getboundingpts, 285
 f_polyhedron24_doubles3_class::getcobject, 292
 f_polyhedron24_doubles3_class::getdata, 296
 f_polyhedron24_doubles3_class::getpt, 302
 f_polyhedron24_doubles3_class::new, 469
 f_polyhedron24_doubles3_class::polyhedron24 ←
 doubles3_type, 499

f_polyhedron24_doubles3_class::setdata, 570
f_polyhedron24_doubles3_class::setpt, 575
f_r2neighborhood_rectangularcuboid_class, 74
f_r2neighborhood_rectangularcuboid_class.f90, 653
f_r2neighborhood_rectangularcuboid_class::F_R2←
 PNeighborhood_RectangularCuboid_add←
 Member, 250
f_r2neighborhood_rectangularcuboid_class::F_R2←
 PNeighborhood_RectangularCuboid_delete,
 250
f_r2neighborhood_rectangularcuboid_class::F_R2P←
 Neighborhood_RectangularCuboid_empty←
 Neighborhood, 251
f_r2neighborhood_rectangularcuboid_class::F_R2P←
 Neighborhood_RectangularCuboid_new, 251
f_r2neighborhood_rectangularcuboid_class::F_R2←
 PNeighborhood_RectangularCuboid_set←
 CenterOfStencil, 251
f_r2neighborhood_rectangularcuboid_class::F_R2←
 PNeighborhood_RectangularCuboid_set←
 Member, 251
f_r2neighborhood_rectangularcuboid_class::F_R2P←
 Neighborhood_RectangularCuboid_setSize,
 252
f_r2neighborhood_rectangularcuboid_class::F_R2←
 PNeighborhood_RectangularCuboid_set←
 SurfaceArea, 252
f_r2neighborhood_rectangularcuboid_class::addmember,
 126
f_r2neighborhood_rectangularcuboid_class::c←
 r2neighborhood_rectangularcuboid, 154
f_r2neighborhood_rectangularcuboid_class::emptyneighborhood, 203
f_r2neighborhood_rectangularcuboid_class::getcobject,
 293
f_r2neighborhood_rectangularcuboid_class::new, 470
f_r2neighborhood_rectangularcuboid_class::r2neighborhood←
 _rectangularcuboid_type, 536
f_r2neighborhood_rectangularcuboid_class::setcenterofstencil, 569
f_r2neighborhood_rectangularcuboid_class::setmember, f_rectangularcuboid_class, 77
f_r2neighborhood_rectangularcuboid_class::setszie, 572
f_r2neighborhood_rectangularcuboid_class::setsurfacearea, f_rectangularcuboid_class::F_RectangularCuboid_←
 construct, 257
f_reconstructioninterface, 75
f_reconstructioninterface::F_reconstructionWith←
 AdvectionNormals_ListedVM_VMAN_RC, 252
f_reconstructioninterface::F_reconstructionWith←
 AdvectionNormalsDebug_ListedVM_VMA←
 N_RC, 252
f_reconstructioninterface::F_reconstructionWithELVI←
 RA2D, 253
f_reconstructioninterface::F_reconstructionWithELVI←
 RA3D, 253
f_reconstructioninterface::F_reconstructionWithLVR←
 A2D_RC, 253
f_reconstructioninterface::F_reconstructionWithLVIR←
 A3D_RC, 253
f_reconstructioninterface::F_reconstructionWithMOF2←
 D_RectangularCuboid, 254
f_reconstructioninterface::F_reconstructionWithMOF2←
 D_Tri, 254
f_reconstructioninterface::F_reconstructionWithMOF2←
 DGiveWeights_RectangularCuboid, 254
f_reconstructioninterface::F_reconstructionWithMOF2←
 DGiveWeights_Tri, 254
f_reconstructioninterface::F_reconstructionWithMOF3←
 D_RectangularCuboid, 255
f_reconstructioninterface::F_reconstructionWithMOF3←
 D_Tet, 255
f_reconstructioninterface::F_reconstructionWithMOF3←
 DGiveWeights_RectangularCuboid, 255
f_reconstructioninterface::F_reconstructionWithMOF3←
 DGiveWeights_Tet, 255
f_reconstructioninterface::F_reconstructionWithR2P2←
 D_RC, 256
f_reconstructioninterface::F_reconstructionWithR2P2←
 DDebug_RC, 256
f_reconstructioninterface::F_reconstructionWithR2P3←
 D_RC, 256
f_reconstructioninterface::F_reconstructionWithR2P3←
 DDebug_RC, 256
f_reconstructioninterface::reconstructionwithadvecednormals,
 538
f_reconstructioninterface::reconstructionwithadvecednormalsdebug,
 539
f_reconstructioninterface::reconstructionwithlvira2d, 539
f_reconstructioninterface::reconstructionwithlvira3d, 539
f_reconstructioninterface::reconstructionwithmof2d, 539
f_reconstructioninterface::reconstructionwithmof3d, 540
f_reconstructioninterface::reconstructionwithr2p2d, 540
f_reconstructioninterface::reconstructionwithr2p2ddebg, 540
f_reconstructioninterface::reconstructionwithr2p3d, 540
f_reconstructioninterface::reconstructionwithr2p3ddebg, 541
f_rectangularcuboid_class, 77
f_rectangularcuboid_class.f90, 654
f_rectangularcuboid_class::F_RectangularCuboid_←
 calculateVolume, 257
f_rectangularcuboid_class::F_RectangularCuboid_←
 construct, 257
f_rectangularcuboid_class::F_RectangularCuboid_←
 construct_2pt, 257
f_rectangularcuboid_class::F_RectangularCuboid_←
 delete, 257
f_rectangularcuboid_class::F_RectangularCuboid_←
 getBoundingPts, 258
f_rectangularcuboid_class::F_RectangularCuboid_new,
 258
f_rectangularcuboid_class::c_rectangularcuboid, 155
f_rectangularcuboid_class::calculatevolume, 165
f_rectangularcuboid_class::construct, 180
f_rectangularcuboid_class::construct_2pt, 181

f_rectangularcuboid_class::getboundingpts, 285
 f_rectangularcuboid_class::getcobject, 294
 f_rectangularcuboid_class::new, 467
 f_rectangularcuboid_class::rectangularcuboid_type,
 541
 f_sepvm_class, 77
 f_sepvm_class.f90, 655
 f_sepvm_class::F_SepVM_construct, 258
 f_sepvm_class::F_SepVM_delete, 258
 f_sepvm_class::F_SepVM_getCentroid, 261
 f_sepvm_class::F_SepVM_getCentroidPtr, 261
 f_sepvm_class::F_SepVM_getVolume, 261
 f_sepvm_class::F_SepVM_getVolumePtr, 262
 f_sepvm_class::F_SepVM_multiplyByVolume, 262
 f_sepvm_class::F_SepVM_new, 262
 f_sepvm_class::F_SepVM_normalizeByVolume, 262
 f_sepvm_class::c_sepvm, 156
 f_sepvm_class::construct, 180
 f_sepvm_class::getcentroid, 286
 f_sepvm_class::getcentroidptr, 287
 f_sepvm_class::getcobject, 294
 f_sepvm_class::getvolume, 306
 f_sepvm_class::getvolumeptr, 316
 f_sepvm_class::multiplybyvolume, 464
 f_sepvm_class::new, 471
 f_sepvm_class::normalizebyvolume, 475
 f_sepvm_class::sepvm_type, 568
 f_sepvm_doubles3_class, 78
 f_sepvm_doubles3_class.f90, 656
 f_sepvm_doubles3_class::F_SepVM_doubles3_delete,
 259
 f_sepvm_doubles3_class::F_SepVM_doubles3_get←
 Centroid, 259
 f_sepvm_doubles3_class::F_SepVM_doubles3_get←
 CentroidPtr, 259
 f_sepvm_doubles3_class::F_SepVM_doubles3_get←
 Data, 259
 f_sepvm_doubles3_class::F_SepVM_doubles3_get←
 Volume, 260
 f_sepvm_doubles3_class::F_SepVM_doubles3_get←
 VolumePtr, 260
 f_sepvm_doubles3_class::F_SepVM_doubles3_get←
 multiplyByVolume, 260
 f_sepvm_doubles3_class::F_SepVM_doubles3_new,
 260
 f_sepvm_doubles3_class::F_SepVM_doubles3_get←
 normalizeByVolume, 261
 f_sepvm_doubles3_class::c_sepvm_doubles3, 157
 f_sepvm_doubles3_class::getcentroid, 286
 f_sepvm_doubles3_class::getcentroidptr, 287
 f_sepvm_doubles3_class::getcobject, 294
 f_sepvm_doubles3_class::getdata, 296
 f_sepvm_doubles3_class::getvolume, 306
 f_sepvm_doubles3_class::getvolumeptr, 316
 f_sepvm_doubles3_class::multiplybyvolume, 464
 f_sepvm_doubles3_class::new, 471
 f_sepvm_doubles3_class::normalizebyvolume, 475
 f_sepvm_doubles3_class::sepvm_doubles3_type, 568
 f_serializer, 79
 f_serializer.f90, 657
 f_serializer::F_Serializer_serializeAndPack_Planar←
 Separator_ByteBuffer, 263
 f_serializer::F_Serializer_unpackAndStore_Planar←
 Separator_ByteBuffer, 263
 f_serializer::serializeandpack, 569
 f_serializer::unpackandstore, 604
 f_tagged_accumlistedvm_vman_class, 80
 f_tagged_accumlistedvm_vman_class.f90, 657
 f_tagged_accumlistedvm_vman_class::F_Tagged_←
 AccumListedVM_VMAN_append, 264
 f_tagged_accumlistedvm_vman_class::F_Tagged_←
 AccumListedVM_VMAN_clear, 264
 f_tagged_accumlistedvm_vman_class::F_Tagged_←
 AccumListedVM_VMAN_delete, 264
 f_tagged_accumlistedvm_vman_class::F_Tagged_←
 AccumListedVM_VMAN_getListAtIndex, 264
 f_tagged_accumlistedvm_vman_class::F_Tagged_←
 AccumListedVM_VMAN_getSize, 265
 f_tagged_accumlistedvm_vman_class::F_Tagged_←
 AccumListedVM_VMAN_getTagForIndex, 265
 f_tagged_accumlistedvm_vman_class::F_Tagged_←
 AccumListedVM_VMAN_new, 265
 f_tagged_accumlistedvm_vman_class::append, 129
 f_tagged_accumlistedvm_vman_class::c_tagged_←
 accumlistedvm_vman, 157
 f_tagged_accumlistedvm_vman_class::clear, 174
 f_tagged_accumlistedvm_vman_class::getcobject, 295
 f_tagged_accumlistedvm_vman_class::getlistatindex,
 297
 f_tagged_accumlistedvm_vman_class::getsize, 304
 f_tagged_accumlistedvm_vman_class::gettagforindex,
 305
 f_tagged_accumlistedvm_vman_class::new, 468
 f_tagged_accumlistedvm_vman_class::tagged_←
 accumlistedvm_vman_type, 592
 f_tagged_accumvm_sepvm_class, 81
 f_tagged_accumvm_sepvm_class.f90, 658
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
 VM_SepVM_delete, 265
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
 VM_SepVM_getCentroidAtIndex, 266
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
 VM_SepVM_getCentroidAtTag, 266
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
 VM_SepVM_getCentroidPtrAtIndex, 266
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
 VM_SepVM_getSize, 266
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
 VM_SepVM_getTagForIndex, 267
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
 VM_SepVM_getVolumeAtIndex, 267
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
 VM_SepVM_getVolumeAtTag, 267
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
 VM_SepVM_getVolumePtrAtIndex, 267
 f_tagged_accumvm_sepvm_class::F_Tagged_Accum←

VM_SepVM_multiplyByVolume, 268
f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
VM_SepVM_new, 268
f_tagged_accumvm_sepvm_class::F_Tagged_Accum←
VM_SepVM_normalizeByVolume, 268
f_tagged_accumvm_sepvm_class::c_tagged_accumvm←
_sepvm, 158
f_tagged_accumvm_sepvm_class::getcentroidatindex,
286
f_tagged_accumvm_sepvm_class::getcentroidattag,
287
f_tagged_accumvm_sepvm_class::getcentroidptratindex,
288
f_tagged_accumvm_sepvm_class::getcobject, 288
f_tagged_accumvm_sepvm_class::getsize, 304
f_tagged_accumvm_sepvm_class::gettagforindex, 305
f_tagged_accumvm_sepvm_class::getvolumeatindex,
306
f_tagged_accumvm_sepvm_class::getvolumeattag, 307
f_tagged_accumvm_sepvm_class::getvolumepratindex,
316
f_tagged_accumvm_sepvm_class::multiplybyvolume,
463
f_tagged_accumvm_sepvm_class::new, 471
f_tagged_accumvm_sepvm_class::normalizebyvolume,
474
f_tagged_accumvm_sepvm_class::tagged_accumvm←
_sepvm_type, 593
f_tagged_accumvm_vm_class, 82
f_tagged_accumvm_vm_class.f90, 660
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_delete, 268
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_getCentroidAtIndex, 269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_getCentroidPtrAtIndex, 269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_getSize, 269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_getTagForIndex, 269
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_getVolumeAtIndex, 270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_getVolumePtrAtIndex, 270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_multiplyByVolume, 270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_new, 270
f_tagged_accumvm_vm_class::F_Tagged_AccumVM←
_VM_normalizeByVolume, 271
f_tagged_accumvm_vm_class::c_tagged_accumvm←
_vm, 158
f_tagged_accumvm_vm_class::getcentroidatindex, 287
f_tagged_accumvm_vm_class::getcentroidptratindex,
288
f_tagged_accumvm_vm_class::getcobject, 289
f_tagged_accumvm_vm_class::getsize, 303
f_tagged_accumvm_vm_class::gettagforindex, 305
f_tagged_accumvm_vm_class::getvolumeatindex, 307
f_tagged_accumvm_vm_class::getvolumepratindex,
316
f_tagged_accumvm_vm_class::multiplybyvolume, 464
f_tagged_accumvm_vm_class::new, 468
f_tagged_accumvm_vm_class::normalizebyvolume,
474
f_tagged_accumvm_vm_class::tagged_accumvm←
_vm_type, 594
f_tet_class, 83
f_tet_class.f90, 661
f_tet_class::F_Tet_construct, 271
f_tet_class::F_Tet_delete, 271
f_tet_class::F_Tet_getBoundingPts, 271
f_tet_class::F_Tet_new, 272
f_tet_class::c_tet, 159
f_tet_class::construct, 178
f_tet_class::getboundingpts, 283
f_tet_class::getcobject, 290
f_tet_class::new, 466
f_tet_class::tet_type, 597
f_tri_class, 84
f_tri_class.f90, 662
f_tri_class::F_Tri_calculateAndSetPlaneOfExistence,
272
f_tri_class::F_Tri_calculateCentroid, 272
f_tri_class::F_Tri_calculateNormal, 272
f_tri_class::F_Tri_calculateSign, 273
f_tri_class::F_Tri_calculateVolume, 273
f_tri_class::F_Tri_construct, 273
f_tri_class::F_Tri_delete, 273
f_tri_class::F_Tri_getBoundingPts, 274
f_tri_class::F_Tri_getLocalizer, 274
f_tri_class::F_Tri_getPlaneOfExistence, 274
f_tri_class::F_Tri_getVertices, 274
f_tri_class::F_Tri_new, 275
f_tri_class::F_Tri_reversePtOrdering, 275
f_tri_class::F_Tri_setPlaneOfExistence, 275
f_tri_class::c_tri, 160
f_tri_class::calculateandsetplaneofexistence, 161
f_tri_class::calculatecentroid, 162
f_tri_class::calculatenormal, 163
f_tri_class::calculatesign, 164
f_tri_class::calculatevolume, 165
f_tri_class::construct, 178
f_tri_class::getboundingpts, 284
f_tri_class::getcobject, 290
f_tri_class::getlocalizer, 297
f_tri_class::getplaneofexistence, 300
f_tri_class::getvertices, 305
f_tri_class::new, 467
f_tri_class::reverseptordering, 547
f_tri_class::setplaneofexistence, 574
f_tri_class::tri_type, 599
f_vman_class, 85
f_vman_class.f90, 663
f_vman_class::F_VMAN_delete, 275
f_vman_class::F_VMAN_getCentroid, 276

f_vman_class::F_VMAN_getNormal, 276
f_vman_class::F_VMAN_getVolume, 276
f_vman_class::F_VMAN_multiplyByVolume, 276
f_vman_class::F_VMAN_new, 277
f_vman_class::F_VMAN_normalizeByVolume, 277
f_vman_class::c_vman, 160
f_vman_class::getcentroid, 286
f_vman_class::getobject, 291
f_vman_class::getnormal, 298
f_vman_class::getvolume, 306
f_vman_class::multiplybyvolume, 464
f_vman_class::new, 467
f_vman_class::normalizebyvolume, 475
f_vman_class::vman_type, 608
f_volumefractionmatching, 85
f_volumefractionmatching.f90, 664
f_volumefractionmatching::F_setDistanceToMatch←
 VolumeFraction_RC_PS_DefTol, 263
f_volumefractionmatching::F_setDistanceToMatch←
 VolumeFraction_RC_PS, 263
f_volumefractionmatching::setdistancetomatchvolumefraction,
 570
fillGeometryAndWeightVectors
 IRL::R2PCommon, 530
fillWithComplementMoments
 IRL::SeparatedMomentsCommon, 566
findDistanceOnePlane
 IRL, 108
fininite_difference_angle_m
 IRL::LVIRACommon, 445
fromEdgeIntersection
 IRL::Pt, 512

getDeltaRotationQuat
 IRL::LVIRA_2D, 438
 IRL::LVIRA_3D, 441
 IRL::R2P_2D1P, 518
 IRL::R2P_2D2P, 521
 IRL::R2P_3D1P, 523
 IRL::R2P_3D2P, 526
getGeometricCasId
 IRL, 109
getPlanePolygonFromReconstruction
 IRL, 109
getReconstructionSurfaceArea
 IRL, 110
getSimplexVerticesAndIntersectionPoints
 IRL::SimplexWrapper< SimplexType, enable_if_t<
 is_tet< SimplexType >::value > >, 579

helper.h, 665

IRL::AccumulateIntoCollection, 122
IRL::AccumulateIntoScalar, 123
IRL::AccumulateWrapper< WrappedType >, 123
IRL::AccumulatedListedVolumeMoments< Volume←
 MomentsType >, 119
IRL::AccumulatedVolumeMoments< VolumeMoments←
 Type >, 121

IRL::Accumulator< ObjectType >, 124
IRL::AdvectionPlaneReconstruction, 127
IRL::AdvectionPlaneReconstructionDebug, 128
IRL::BFGS< OptimizingClass, kParameters >, 130
IRL::Bisection
 solve, 132
IRL::Bisection< OptimizingClass >, 131
IRL::BlockObjectAllocation< ObjectType >, 133
IRL::BrentsMethod
 solve, 135
IRL::BrentsMethod< OptimizingClass >, 134
IRL::ByteBuffer, 136
IRL::CappedDodecahedronSpecialization< Derived,
 VertexType >, 167
IRL::CellCollection< CellGroupType >, 168
IRL::CellGroupedMoments< CellType, Contained←
 MomentsType >, 169
IRL::Centroid2D_Functor, 171
IRL::Centroid3D_Functor, 172
IRL::ChainedBlockStorage< ContainedType, kBlock←
 Size >, 173
IRL::Collection< ObjectType >, 174
IRL::ConcaveBoxSpecialization< Derived, VertexType
 >, 176
IRL::ConstIteratorThroughBracketOperator< Container←
 Type >, 177
IRL::ContinueDividingVolumeByPlanarReconstruction<
 HowToCutSimplexByNextPlane, HowTo←
 ToBranchEarlyIfNoIntersection, HowTo←
 GenerateAndHandleSimplicesFromVolume←
 UnderPlane, HowToGenerateAndHandle←
 SimplicesFromVolumeAbovePlane >, 181
IRL::ContinueReducingVolumeToBeInternalToReconstruction,
 182
IRL::CutSimplexByNextPlaneAccessedByIndex, 182
IRL::CutTetVertices< VertexType >, 182
IRL::CutTriangleVerticesAndPlane< VertexType >, 183
IRL::DecomposedPolygonVertexStorage< VertexType
 >, 184
IRL::DecomposedPolyhedronVertexStorage< Vertex←
 Type >, 186
IRL::DecomposedPolytopeVertexStorage< VertexType
 >, 187
IRL::DelayedExpandableVertexAccess< Derived,
 VertexType, kMaskedVertices >, 188
IRL::DivideSimplexByPlanarReconstruction< HowTo←
 HandleVolumeCompletelyInternalToPlanar←
 Reconstruction, HowToContinueDividing←
 VolumeByPlanarReconstruction >, 191
IRL::DividedPolygonSpecialization< Derived, Vertex←
 Type >, 189
IRL::DodecahedronSpecialization< Derived, Vertex←
 Type >, 192
IRL::DoesNotHaveACollection< ReturnType >, 193
IRL::DoesNotHaveALocalizer< ReconstructionType >,
 193
IRL::DoesNotHaveANestedType< ReturnType >, 194
IRL::DoesNotHaveAReconstructionLink

value, 194
IRL::DoesNotHaveAReconstructionLink< ReconstructionType >, 194
IRL::DoesNotHaveASeparator< ReconstructionType >, 194
IRL::ELVIRA_2D, 196
IRL::ELVIRA_3D, 197
IRL::ELVIRADebug< BaseELVIRA >, 199
IRL::ELVIRADebug< ELVIRA_2D >, 199
IRL::ELVIRADebug< ELVIRA_3D >, 200
IRL::ELVIRANeighborhood, 201
IRL::ExpandableDividedPolygon< VertexType >, 204
IRL::ExpandablePolygon< VertexType >, 205
IRL::ExpandablePtList< VertexType >, 206
IRL::Expr< E >, 208
IRL::Face< HalfEdgeType >, 277
IRL::GeneralPolygon< Derived, VertexType, SimplexType >, 279
IRL::GeneralPolyhedron< Derived, VertexType, SimplexType >, 281
IRL::HalfEdge< VertexType >, 317
IRL::HalfEdgeCutting, 318
IRL::HalfEdgePolygon< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >, 318
IRL::HalfEdgePolyhedron< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >, 320
IRL::HalfEdgePolytope< PtType, VertexLocationType, VertexType, HalfEdgeType, FaceType, kMaxHalfEdges, kMaxVertices, kMaxFaces >, 321
IRL::HasACollection< ReturnType >, 347
IRL::HasALocalizer< ReconstructionType >, 347
IRL::HasANestedType< ReturnType >, 348
IRL::HasAReconstructionLink
 value, 348
IRL::HasAReconstructionLink< ReconstructionType >, 348
IRL::HasASeparator< ReconstructionType >, 348
IRL::HasLocalizer_AND_HasSeparator
 value, 349
IRL::HasLocalizer_AND_HasSeparator< ReconstructionType >, 349
IRL::HasLocalizer_OR_HasSeparator
 value, 349
IRL::HasLocalizer_OR_HasSeparator< ReconstructionType >, 349
IRL::HasLocalizer_XOR_HasSeparator
 value, 350
IRL::HasLocalizer_XOR_HasSeparator< ReconstructionType >, 350
IRL::HexahedronSpecialization< Derived, VertexType >, 350
IRL::IgnoreVolumeAbovePlane, 351
IRL::Illinois
 solve, 352, 353
 IRL::Illinois< OptimizingClass >, 351
 IRL::IsNotNullReconstruction
 value, 410
 IRL::IsNotNullReconstruction< ReconstructionType >, 410
 IRL::IsNotAPlanarSeparator< ReconstructionType >, 411
 IRL::IsNullReconstruction
 value, 411
 IRL::IsNullReconstruction< ReconstructionType >, 411
 IRL::IsPlanarSeparator< ReconstructionType >, 412
 IRL::IterativeSolverForDistance
 IterativeSolverForDistance, 420, 421
 solve, 421
 IRL::IterativeSolverForDistance< CellType, kMaxPlanes >, 418
 IRL::IteratorThroughBracketOperator< ContainerType >, 422
 IRL::JoinedReconstructions< CurrentReconstruction, NextReconstruction >, 423
 IRL::JoinedReconstructions< CurrentReconstruction, NullReconstruction >, 424
 IRL::JoinedReconstructionsToGroup< CurrentReconstruction, GroupReconstruction >, 425
 IRL::KMeans, 425
 IRL::LVIRA_2D< CellType >, 436
 IRL::LVIRA_2D
 getDeltaRotationQuat, 438
 IRL::LVIRA_3D< CellType >, 439
 IRL::LVIRA_3D
 getDeltaRotationQuat, 441
 IRL::LVIRACommon
 finite_difference_angle_m, 445
 runOptimization, 444
 IRL::LVIRACommon< CellType, kColumns >, 442
 IRL::LVIRADebug< LVIRAType >, 446
 IRL::LVIRANeighborhood< CellType >, 447
 IRL::LevenbergMarquardt
 reason_for_exit_m, 428
 IRL::LevenbergMarquardt< OptimizingClass, -1, kColumns >, 429
 reason_for_exit_m, 430
 IRL::LevenbergMarquardt< OptimizingClass, kRows, kColumns >, 426
 IRL::LinearInterpolation_Functor, 430
 IRL::ListedVolumeMoments< VolumeMomentsType >, 431
 IRL::Lister< ObjectType >, 433
 IRL::LocalizerLinkFromLocalizedSeparatorLink, 435
 IRL::MOF_2D< CellType >, 453
 IRL::MOF_3D< CellType >, 455
 IRL::MOFCommon
 initial_angle_m, 458
 IRL::MOFCommon< CellType >, 456
 IRL::MOFDebug< MOFType >, 459
 IRL::MaskStripper< MaskedType >, 451
 IRL::MaskedLocalizedSeparatorLink, 450

IRL::MathVector< kNumberOfElements >, 452
 IRL::MomentCalculationType< VertexType, ProxyType, Enable >, 461
 IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polygon< ProxyType >::value > >, 461
 IRL::MomentCalculationType< VertexType, ProxyType, enable_if_t< is_polyhedron< ProxyType >::value > >, 461
 IRL::MultiMaterialAccumulatedVolumeMoments< MomentsType >, 462
 IRL::Normal, 472
 IRL::NullReconstruction, 475
 IRL::ObjectAllocationServer< ObjectType >, 476
 IRL::PartitionByNormal< MomentsContainerType >, 478
 IRL::PassToNestedType, 480
 IRL::PathGraphNode< NodeType >, 480
 IRL::PlanarLocalizer, 482
 IRL::PlanarReconstruction< kStackPlanes >, 485
 IRL::PlanarSeparator, 487
 IRL::PlanarSeparatorPathGroup, 490
 IRL::Plane, 491
 IRL::PolygonBase< VertexType >, 493
 IRL::PolygonConstructors< Derived, VertexType, StorageType, PlaneOfExistenceType >, 493
 IRL::PolygonMomentsCalculation< Derived, VertexType, SimplexType >, 494
 IRL::PolygonMomentsCalculationCommon< Derived, VertexType, SimplexType >, 495
 IRL::PolygonSpecialization< Derived, VertexType >, 496
 IRL::Polyhedron< VertexType, kMaxNumberOfPts >, 499
 IRL::Polyhedron24Specialization< Derived, VertexType >, 500
 IRL::PolyhedronMomentsCalculation< Derived, PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, SimplexType >, 502
 IRL::PolyhedronMomentsCalculation< Derived, VertexType, SimplexType >, 501
 IRL::PolyhedronMomentsCalculationCommon< Derived, VertexType, SimplexType >, 503
 IRL::ProxyPolygonConstructors< Derived, GeometryType, kNumberOfVertices, PlaneOfExistenceType >, 505
 IRL::ProxyTet< GeometryType >, 506
 IRL::ProxyTri< GeometryType >, 507
 IRL::ProxyVertexAccess< Derived, GeometryType, kNumberOfVertices >, 508
 IRL::Pt, 509
 fromEdgeIntersection, 512
 IRL::PtList< VertexType, kMaxNumberOfPts >, 513
 IRL::PtWithDataCommon< Derived, AttachedDataType >, 514
 IRL::PtWithDoublesStatelessFunctor< FunctorType, kArrayLength >, 515
 IRL::R2P_2D1P< CellType >, 516
 IRL::R2P_2D1P
 getDeltaRotationQuat, 518
 updateGuess, 518
 IRL::R2P_2D2P< CellType >, 519
 IRL::R2P_2D2P
 getDeltaRotationQuat, 521
 updateGuess, 521
 IRL::R2P_3D1P< CellType >, 521
 IRL::R2P_3D1P
 getDeltaRotationQuat, 523
 updateGuess, 523
 IRL::R2P_3D2P< CellType >, 524
 IRL::R2P_3D2P
 getDeltaRotationQuat, 526
 updateGuess, 526
 IRL::R2PCommon
 addCellMomentsToGeometryVector, 530
 fillGeometryAndWeightVectors, 530
 initial_angle_m, 532
 runOptimization, 531
 setRelativeImportanceBetweenWeights, 531
 IRL::R2PCommon< CellType, kColumns >, 527
 IRL::R2PDebug< R2PType >, 532
 IRL::R2PNeighborhood< CellType >, 534
 IRL::ReconstructionLink< ReconstructionType, GraphType >, 537
 IRL::RectangularCuboidBase< VertexType >, 541
 IRL::RectangularCuboidCommon< Derived, VertexType >, 542
 IRL::RectangularCuboidSpecialization< Derived, VertexType >, 542
 IRL::RecursiveSimplexCutting, 543
 IRL::ReferenceFrame, 543
 IRL::ReferredToPlaneOfExistence, 544
 IRL::RestrictedLocalizerLinkFromLocalizedSeparatorLink, 545
 IRL::RestrictedMaskedLocalizedSeparatorLink, 546
 IRL::Secant
 reason_for_exit_m, 550
 solve, 549
 solveAndTrackBounds, 550
 IRL::Secant< OptimizingClass >, 548
 IRL::SegmentedDecomposedPolygon< VertexType >, 551
 IRL::SegmentedDecomposedPolyhedron< VertexType >, 552
 IRL::SegmentedDecomposedPolytope< VertexType, ProxyType >, 553
 IRL::SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices >, 554
 IRL::SegmentedHalfEdgePolygonCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >, 555
 IRL::SegmentedHalfEdgePolygonSpecificPt< Pt, FaceType, VertexType, kMaxFaces, kMaxVertices >, 556
 IRL::SegmentedHalfEdgePolygonSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices >, 556

IRL::SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices >, 556
IRL::SegmentedHalfEdgePolyhedronCommon< FaceType, VertexType, kMaxFaces, kMaxVertices >, 557
IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices >, 558
IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtType, FaceType, VertexType, kMaxFaces, kMaxVertices >, 559
IRL::SegmentedHalfEdgePolyhedronSpecificPt< PtType, WithDoublesStatelessFunctor< FunctorType, kArrayLength >, FaceType, VertexType, kMaxFaces, kMaxVertices >, 559
IRL::SegmentedHalfEdgePolytope< FaceType, VertexType, kMaxFaces, kMaxVertices >, 559
IRL::SelfExpandingCollection< ObjectType >, 561
IRL::SeparatedMoments< MomentsType >, 562
IRL::SeparatedMoments< VolumeMoments >, 562
IRL::SeparatedMoments< VolumeMomentsAndDoubles < kArrayLength > >, 563
IRL::SeparatedMomentsCommon
 fillWithComplementMoments, 566
 SeparatedMomentsCommon, 566
IRL::SeparatedMomentsCommon< Derived, MomentsType >, 564
IRL::ShareVolumeAbovePlaneWithLinkedNeighbor, 577
IRL::SimplexCutting, 578
IRL::SimplexWrapper< SimplexType, Enable >, 578
IRL::SimplexWrapper< SimplexType, enable_if_t< is_tet< SimplexType >::value > >, 578
 getSimplexVerticesAndIntersectionPoints, 579
 simplexFromCutSimplexVertices, 579
IRL::SimplexWrapper< SimplexType, enable_if_t< is_tri< SimplexType >::value > >, 580
IRL::SmallVector< T, BuffSizeInElements >, 580
IRL::StackVector< ObjectType, kMaxSize >, 582
IRL::StolenGraph< NodeType, StolenGraphType >, 583
IRL::StoredCappedDodecahedron< VertexType >, 584
IRL::StoredConcaveBox< VertexType >, 585
IRL::StoredDodecahedron< VertexType >, 585
IRL::StoredHexahedron< VertexType >, 586
IRL::StoredPlaneOfExistence, 586
IRL::StoredPolyhedron24< VertexType >, 587
IRL::StoredRectangularCuboid< VertexType >, 588
IRL::StoredTet< VertexType >, 588
IRL::StoredTri< VertexType >, 589
IRL::StoredVertexAccess< Derived, VertexType, kNumberOfVertices >, 590
IRL::StrongWolfeConditionLineSearch, 591
IRL::TaggedAccumulatedListedVolumeMoments< VolumeMomentsType >, 594
IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType >, 595
IRL::TaggedAccumulatedVolumeMoments< VolumeMomentsType >::TaggedMoments, 597
IRL::TetSpecialization< Derived, VertexType >, 598
IRL::TriSpecialization< Derived, VertexType >, 600
IRL::UnDirectedGraphNode< NodeType >, 602
IRL::UnitQuaternion, 603
IRL::Vertex< VertexLocationType >, 606
IRL::VertexList< VertexStorage, VertexType >, 607
IRL::VertexLocation< PtType >, 608
IRL::Volume, 609
IRL::Volume2D_Functor, 610
IRL::Volume3D_Functor, 611
IRL::VolumeMoments, 611
 VolumeMoments, 613
IRL::VolumeMoments2D_Functor, 614
IRL::VolumeMoments3D_Functor, 615
IRL::VolumeMomentsAndDoubles
 VolumeMomentsAndDoubles, 617
IRL::VolumeMomentsAndDoubles< kArrayLength >, 616
IRL::VolumeMomentsAndDoubles3D_Functor< kArrayLength >, 618
IRL::VolumeMomentsAndNormal, 619
IRL::VolumeMomentsAndNormal2D_Functor, 621
IRL::doNoEarlyBranching, 195
IRL::earlyBranchBothFullyBelowAndFullyAbove< SimplicesLink, 195
IRL::earlyBranchBothFullyBelowAndFullyAbove< SimplicesNoLink, 195
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, NullReconstruction, enable_if_t< IsNullReconstruction >::value > >, 308
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, PlanarSeparator, enable_if_t< is_separated_moments < ReturnType >::value > >, 309
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, Enable >, 307
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isHalfEdgeCutting < CuttingMethod >::value && !IsNotNullReconstruction < ReconstructionType >::value && !(IsPlanar < Separator < ReconstructionType >::value && is_separated_moments < ReturnType >::value) > >, 310
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isRecursiveSimplexCutting < CuttingMethod >::value && !IsNotNullReconstruction < ReconstructionType >::value && !(IsPlanar < Separator < ReconstructionType >::value && is_separated_moments < ReturnType >::value) > >, 310
IRL::generic_cutting_details::getVolumeMoments< ReturnType, CuttingMethod, EncompassingType, ReconstructionType, enable_if_t< isRecursiveSimplexCutting < CuttingMethod >::value && !IsNotNullReconstruction < ReconstructionType >::value && !(IsPlanar < Separator < ReconstructionType >::value && is_separated_moments < ReturnType >::value) > >, 310

```

_t<    isSimplexCutting<    CuttingMethod
>::value    &&!IsNotNullReconstruction<
ReconstructionType >::value &&!IsPlanar<
Separator< ReconstructionType >::value
&&is_separated_moments< ReturnType >::value> >, 311
IRL::generic_cutting_details::getVolumeMoments<
ProvidedStorage< ReturnType, Cutting<
Method, SegmentedPolytopeType, Half<
EdgePolytopeType, NullReconstruction,
enable_if_t< IsNullReconstruction< Null<
Reconstruction >::value > >, 312
IRL::generic_cutting_details::getVolumeMoments<
ProvidedStorage< ReturnType, Cutting<
Method, SegmentedPolytopeType, Half<
EdgePolytopeType, PlanarSeparator, enable<
_if_t< isHalfEdgeCutting< CuttingMethod
>::value &&is_separated_moments< ReturnType >::value > >, 312
IRL::generic_cutting_details::getVolumeMoments<
ProvidedStorage< ReturnType, Cutting<
Method, SegmentedPolytopeType, Half<
EdgePolytopeType, PlanarSeparator, enable<
_if_t< isSimplexCutting< CuttingMethod >::value &&is_separated_moments< ReturnType >::value > >, 314
IRL::generic_cutting_details::getVolumeMoments<
ProvidedStorage< ReturnType, Cutting<
Method, SegmentedPolytopeType, Half<
EdgePolytopeType, ReconstructionType, Enable >, 311
IRL::generic_cutting_details::getVolumeMoments<
ProvidedStorage< ReturnType, Cutting<
Method, SegmentedPolytopeType, Half<
EdgePolytopeType, ReconstructionType, enable_if_t< isHalfEdgeCutting< Cutting<
Method >::value &&!IsNotNullReconstruction< ReconstructionType >::value &&!IsPlanar<
Separator< ReconstructionType >::value
&&is_separated_moments< ReturnType >::value> >, 315
IRL::generic_cutting_details::getVolumeMoments<
ProvidedStorage< ReturnType, Cutting<
Method, SegmentedPolytopeType, Half<
EdgePolytopeType, ReconstructionType, enable_if_t< isSimplexCutting< Cutting<
Method >::value &&!IsNotNullReconstruction< ReconstructionType >::value &&!IsPlanar<
Separator< ReconstructionType >::value
&&is_separated_moments< ReturnType >::value> >, 315
IRL::has_localizer< C >, 322
IRL::has_localizer< const C >, 323
IRL::has_localizer< LocalizedSeparator >, 324
IRL::has_localizer< LocalizedSeparatorGroup >, 325
IRL::has_localizer< LocalizedSeparatorGroupLink >, 326
IRL::has_localizer< LocalizedSeparatorLink >, 327
IRL::has_localizer< LocalizerLink >, 328
IRL::has_localizer< LocalizerLinkFromLocalized<
SeparatorLink >, 329
IRL::has_localizer< MaskedLocalizedSeparatorLink >, 330
IRL::has_localizer< PlanarLocalizer >, 331
IRL::has_localizer< RestrictedLocalizerLinkFrom<
LocalizedSeparatorLink >, 332
IRL::has_localizer< RestrictedMaskedLocalized<
SeparatorLink >, 333
IRL::has_null_reconstruction< C >, 334
IRL::has_null_reconstruction< const C >, 335
IRL::has_null_reconstruction< NullReconstruction >, 336
IRL::has_separator< C >, 337
IRL::has_separator< const C >, 338
IRL::has_separator< LocalizedSeparator >, 339
IRL::has_separator< LocalizedSeparatorGroup >, 340
IRL::has_separator< LocalizedSeparatorGroupLink >, 341
IRL::has_separator< LocalizedSeparatorLink >, 342
IRL::has_separator< MaskedLocalizedSeparatorLink >, 343
IRL::has_separator< PlanarSeparator >, 344
IRL::has_separator< PlanarSeparatorPath >, 345
IRL::has_separator< RestrictedMaskedLocalized<
SeparatorLink >, 346
IRL::is_moments_collection< AccumulatedListed<
VolumeMoments< VolumeMomentsType > >, 355
IRL::is_moments_collection< AccumulatedVolume<
Moments< VolumeMomentsType > >, 356
IRL::is_moments_collection< C >, 353
IRL::is_moments_collection< const C >, 357
IRL::is_moments_collection< ListedVolumeMoments<
VolumeMomentsType > >, 358
IRL::is_moments_collection< TaggedAccumulated<
ListedVolumeMoments< VolumeMoments< VolumeMomentsType > >, 359
IRL::is_moments_collection< TaggedAccumulated<
VolumeMoments< VolumeMomentsType > >, 360
IRL::is_nested_moments< AccumulateWrapper<
NestedType > >, 362
IRL::is_nested_moments< C >, 360
IRL::is_nested_moments< const C >, 363
IRL::is_planar_separator< C >, 364
IRL::is_planar_separator< const C >, 365
IRL::is_planar_separator< PlanarSeparator >, 366
IRL::is_polygon< C >, 367
IRL::is_polygon< const C >, 368
IRL::is_polygon< ExpandableDividedPolygon< Vertex<
Type > > >, 369
IRL::is_polygon< ExpandablePolygon< VertexType > >, 370
IRL::is_polygon< ProxyTri< GeometryType > >, 371
IRL::is_polygon< SegmentedDecomposedPolygon<
VertexType > > >, 372

```

IRL::is_polygon< SegmentedHalfEdgePolygon< FaceType, VertexType, kMaxFaces, kMaxVertices > >, 373
IRL::is_polygon< StoredTri< VertexType > >, 374
IRL::is_polyhedron< C >, 375
IRL::is_polyhedron< const C >, 376
IRL::is_polyhedron< ProxyTet< GeometryType > >, 377
IRL::is_polyhedron< SegmentedDecomposed< Polyhedron< VertexType > >, 378
IRL::is_polyhedron< SegmentedHalfEdgePolyhedron< FaceType, VertexType, kMaxFaces, kMaxVertices > >, 379
IRL::is_polyhedron< StoredCappedDodecahedron< VertexType > >, 380
IRL::is_polyhedron< StoredConcaveBox< VertexType > >, 381
IRL::is_polyhedron< StoredDodecahedron< VertexType > >, 382
IRL::is_polyhedron< StoredHexahedron< VertexType > >, 383
IRL::is_polyhedron< StoredPolyhedron24< VertexType > >, 384
IRL::is_polyhedron< StoredRectangularCuboid< VertexType > >, 385
IRL::is_polyhedron< StoredTet< VertexType > >, 386
IRL::reconstruction_link< C >, 387
IRL::reconstruction_link< const C >, 388
IRL::reconstruction_link< LocalizedSeparator< GroupLink >, 389
IRL::reconstruction_link< LocalizedSeparatorLink >, 390
IRL::reconstruction_link< LocalizerLink >, 391
IRL::reconstruction_link< LocalizerLinkFrom< LocalizedSeparatorLink >, 392
IRL::reconstruction_link< MaskedLocalized< SeparatorLink >, 393
IRL::reconstruction_link< PlanarSeparatorPath >, 394
IRL::reconstruction_link< RestrictedLocalizerLink< FromLocalizedSeparatorLink >, 395
IRL::reconstruction_link< RestrictedMasked< LocalizedSeparatorLink >, 396
IRL::separated_moments< C >, 396
IRL::separated_moments< const C >, 398
IRL::separated_moments< SeparatedMoments< MomentsType > >, 399
IRL::is_tet< C >, 400
IRL::is_tet< const C >, 401
IRL::is_tet< ProxyTet< GeometryType > >, 402
IRL::is_tet< StoredTet< VertexType > >, 403
IRL::is_tri< C >, 404
IRL::is_tri< const C >, 405
IRL::is_tri< ProxyTri< GeometryType > >, 406
IRL::is_tri< StoredTri< VertexType > >, 407
IRL::isHalfEdgeCutting< C >, 408
IRL::isHalfEdgeCutting< const C >, 409
IRL::isHalfEdgeCutting< HalfEdgeCutting >, 410
IRL::isRecursiveSimplexCutting< C >, 412
IRL::isRecursiveSimplexCutting< const C >, 413
IRL::isRecursiveSimplexCutting< RecursiveSimplexCutting >, 414
IRL::isSimplexCutting< C >, 415
IRL::isSimplexCutting< const C >, 416
IRL::isSimplexCutting< SimplexCutting >, 417
IRL::vector_sum< V1, V2 >, 604
IRL, 86
 __attribute__, 106
 C_CUTTING_METHOD, 117
 checkPureObject, 106
 crossProduct, 106
 crossProductNormalized, 107
 cutPlaneByHexahedron, 107
 cutPolygonByPlane, 108
 dotProduct, 108
 findDistanceOnePlane, 108
 getGeometricCaseId, 109
 getPlanePolygonFromReconstruction, 109
 getReconstructionSurfaceArea, 110
 isPlaneIntersectingCell, 110
 isPtInternal, 111
 magnitude, 111
 rotateNormalOntoNormal, 112
 scalarTripleProduct, 112
 setDistanceToMatchVolumeFractionPartialFill, 113
 setVolumeFractionBounds, 113
 setVolumeFractionTolerance, 113
 signedDistanceToPolygonVertices, 114
 signedDistanceToVertices, 114
 sortAscendingBasedOnOtherArray, 115
 sortDescendingBasedOnOtherArray, 116
 squaredMagnitude, 116
 unit_cell, 117
initial_angle_m
 IRL::MOFCommon, 458
 IRL::R2PCommon, 532
irl_fortran_interface, 117
irl_fortran_interface.f90, 666
isPlaneIntersectingCell
 IRL, 110
isPtInternal
 IRL, 111
IterativeSolverForDistance
 IRL::IterativeSolverForDistance, 420, 421
lookup_tables.h, 667
 cuboid_cut_vertices, 669
 cut_vertices, 669, 670
 number_of_negative_tets_after_cut, 670
 number_of_negative_tris_after_cut, 671
 number_of_new_vertices_after_cut, 671
 number_of_plane_intersections_with_cuboid, 672
 number_of_tets_after_cut, 672
 number_of_tets_under_plane, 673
 number_of_tris_after_cut, 673
 tet_vertices, 673
 total_number_of_tets, 674

verts_for_tets, 674
 verts_for_tris, 674
 lvira_optimization.h, 675
 MAX_PLANES_IN_OBJECT
 constants.h, 633
 magnitude
 IRL, 111
 mymath.h, 677
 number_of_negative_tets_after_cut
 lookup_tables.h, 670
 number_of_negative_tris_after_cut
 lookup_tables.h, 671
 number_of_new_vertices_after_cut
 lookup_tables.h, 671
 number_of_plane_intersections_with_cuboid
 lookup_tables.h, 672
 number_of_tets_after_cut
 lookup_tables.h, 672
 number_of_tets_under_plane
 lookup_tables.h, 673
 number_of_tris_after_cut
 lookup_tables.h, 673
 plane_distance.h, 678
 r2p_optimization.h, 680
 reason_for_exit_m
 IRL::LevenbergMarquardt, 428
 IRL::LevenbergMarquardt< OptimizingClass, -1,
 kColumns >, 430
 IRL::Secant, 550
 rotateNormalOntoNormal
 IRL, 112
 rotations.h, 682
 runOptimization
 IRL::LVIRACommon, 444
 IRL::R2PCommon, 531
 scalarTripleProduct
 IRL, 112
 SeparatedMomentsCommon
 IRL::SeparatedMomentsCommon, 566
 setDistanceToMatchVolumeFractionPartialFill
 IRL, 113
 setRelativeImportanceBetweenWeights
 IRL::R2PCommon, 531
 setVolumeFractionBounds
 IRL, 113
 setVolumeFractionTolerance
 IRL, 113
 short_alloc::arena< N, alignment >, 129
 short_alloc::short_alloc< T, N, Align >, 577
 short_alloc::short_alloc< T, N, Align >::rebind< _Up >,
 537
 signedDistanceToPolygonVertices
 IRL, 114
 signedDistanceToVertices

IRL, 114
 simplexFromCutSimplexVertices
 IRL::SimplexWrapper< SimplexType, enable_if_t<
 is_tet< SimplexType >::value > >, 579
 solve
 IRL::Bisection, 132
 IRL::BrentsMethod, 135
 IRL::Illinois, 352, 353
 IRL::IterativeSolverForDistance, 421
 IRL::Secant, 549
 solveAndTrackBounds
 IRL::Secant, 550
 sortAscendingBasedOnOtherArray
 IRL, 115
 sortDescendingBasedOnOtherArray
 IRL, 116
 squaredMagnitude
 IRL, 116
 tet_vertices
 lookup_tables.h, 673
 total_number_of_tets
 lookup_tables.h, 674
 unit_cell
 IRL, 117
 updateGuess
 IRL::R2P_2D1P, 518
 IRL::R2P_2D2P, 521
 IRL::R2P_3D1P, 523
 IRL::R2P_3D2P, 526
 value
 IRL::DoesNotHaveAReconstructionLink, 194
 IRL::HasAReconstructionLink, 348
 IRL::HasLocalizer_AND_HasSeparator, 349
 IRL::HasLocalizer_OR_HasSeparator, 349
 IRL::HasLocalizer_XOR_HasSeparator, 350
 IRL::IsNotNullReconstruction, 410
 IRL::IsNullReconstruction, 411
 verts_for_tets
 lookup_tables.h, 674
 verts_for_tris
 lookup_tables.h, 674
 VolumeMoments
 IRL::VolumeMoments, 613
 VolumeMomentsAndDoubles
 IRL::VolumeMomentsAndDoubles, 617