

MkniX  
0.9.0

Generated by Doxygen 1.8.9.1

Sat Dec 5 2015 00:24:54

## Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>1</b>
2.1	Class Hierarchy . . . . .	1
<b>3</b>	<b>Class Index</b>	<b>4</b>
3.1	Class List . . . . .	4
<b>4</b>	<b>File Index</b>	<b>6</b>
4.1	File List . . . . .	6
<b>5</b>	<b>Namespace Documentation</b>	<b>11</b>
5.1	Imx Namespace Reference . . . . .	11
5.2	mnkix Namespace Reference . . . . .	11
5.2.1	Detailed Description . . . . .	12
5.2.2	Typedef Documentation . . . . .	12
5.2.3	Function Documentation . . . . .	13
<b>6</b>	<b>Class Documentation</b>	<b>13</b>
6.1	mnkix::Analysis Class Reference . . . . .	13
6.1.1	Detailed Description . . . . .	15
6.1.2	Constructor & Destructor Documentation . . . . .	15
6.1.3	Member Function Documentation . . . . .	15
6.1.4	Member Data Documentation . . . . .	16
6.2	mnkix::AnalysisDynamic Class Reference . . . . .	17
6.2.1	Detailed Description . . . . .	18
6.2.2	Constructor & Destructor Documentation . . . . .	18
6.2.3	Member Function Documentation . . . . .	18
6.3	mnkix::AnalysisStatic Class Reference . . . . .	19
6.3.1	Detailed Description . . . . .	20
6.3.2	Constructor & Destructor Documentation . . . . .	20
6.3.3	Member Function Documentation . . . . .	21
6.4	mnkix::AnalysisThermalDynamic Class Reference . . . . .	21
6.4.1	Detailed Description . . . . .	22
6.4.2	Constructor & Destructor Documentation . . . . .	22
6.4.3	Member Function Documentation . . . . .	23
6.5	mnkix::AnalysisThermalStatic Class Reference . . . . .	24
6.5.1	Detailed Description . . . . .	24
6.5.2	Constructor & Destructor Documentation . . . . .	25

6.5.3	Member Function Documentation	25
6.6	<a href="#">mknix::AnalysisThermoMechanicalDynamic Class Reference</a>	25
6.6.1	Detailed Description	26
6.6.2	Constructor & Destructor Documentation	27
6.6.3	Member Function Documentation	27
6.7	<a href="#">mknix::Body Class Reference</a>	28
6.7.1	Detailed Description	30
6.7.2	Constructor & Destructor Documentation	30
6.7.3	Member Function Documentation	31
6.7.4	Member Data Documentation	37
6.8	<a href="#">mknix::BoundaryGroup Class Reference</a>	38
6.8.1	Detailed Description	38
6.8.2	Constructor & Destructor Documentation	39
6.8.3	Member Function Documentation	39
6.8.4	Member Data Documentation	40
6.9	<a href="#">mknix::Cell Class Reference</a>	40
6.9.1	Detailed Description	42
6.9.2	Constructor & Destructor Documentation	42
6.9.3	Member Function Documentation	42
6.9.4	Member Data Documentation	44
6.10	<a href="#">mknix::CellBoundary Class Reference</a>	45
6.10.1	Detailed Description	45
6.10.2	Constructor & Destructor Documentation	46
6.10.3	Member Function Documentation	46
6.10.4	Member Data Documentation	47
6.11	<a href="#">mknix::CellBoundaryLinear Class Reference</a>	47
6.11.1	Detailed Description	48
6.11.2	Constructor & Destructor Documentation	49
6.11.3	Member Function Documentation	50
6.11.4	Member Data Documentation	51
6.12	<a href="#">mknix::CellRect Class Reference</a>	51
6.12.1	Detailed Description	52
6.12.2	Constructor & Destructor Documentation	52
6.12.3	Member Function Documentation	52
6.13	<a href="#">mknix::CellTetrahedron Class Reference</a>	53
6.13.1	Detailed Description	54
6.13.2	Constructor & Destructor Documentation	54
6.13.3	Member Function Documentation	55
6.13.4	Member Data Documentation	55
6.14	<a href="#">mknix::CellTriang Class Reference</a>	55

6.14.1 Detailed Description . . . . .	57
6.14.2 Constructor & Destructor Documentation . . . . .	57
6.14.3 Member Function Documentation . . . . .	58
6.14.4 Member Data Documentation . . . . .	58
6.15 mknix::CompBar Class Reference . . . . .	59
6.15.1 Detailed Description . . . . .	59
6.15.2 Constructor & Destructor Documentation . . . . .	59
6.15.3 Member Function Documentation . . . . .	59
6.16 mknix::Constraint Class Reference . . . . .	59
6.16.1 Detailed Description . . . . .	61
6.16.2 Constructor & Destructor Documentation . . . . .	61
6.16.3 Member Function Documentation . . . . .	61
6.16.4 Member Data Documentation . . . . .	64
6.17 mknix::ConstraintClearance Class Reference . . . . .	65
6.17.1 Detailed Description . . . . .	66
6.17.2 Constructor & Destructor Documentation . . . . .	66
6.17.3 Member Function Documentation . . . . .	67
6.17.4 Member Data Documentation . . . . .	67
6.18 mknix::ConstraintContact Class Reference . . . . .	67
6.18.1 Detailed Description . . . . .	69
6.18.2 Constructor & Destructor Documentation . . . . .	69
6.18.3 Member Function Documentation . . . . .	69
6.18.4 Member Data Documentation . . . . .	69
6.19 mknix::ConstraintDistance Class Reference . . . . .	70
6.19.1 Detailed Description . . . . .	71
6.19.2 Constructor & Destructor Documentation . . . . .	71
6.19.3 Member Function Documentation . . . . .	72
6.19.4 Member Data Documentation . . . . .	73
6.20 mknix::ConstraintFixedAxis Class Reference . . . . .	73
6.20.1 Detailed Description . . . . .	74
6.20.2 Constructor & Destructor Documentation . . . . .	74
6.20.3 Member Function Documentation . . . . .	75
6.20.4 Member Data Documentation . . . . .	75
6.21 mknix::ConstraintFixedCoordinates Class Reference . . . . .	76
6.21.1 Detailed Description . . . . .	77
6.21.2 Constructor & Destructor Documentation . . . . .	77
6.21.3 Member Function Documentation . . . . .	77
6.21.4 Member Data Documentation . . . . .	78
6.22 mknix::ConstraintThermal Class Reference . . . . .	78
6.22.1 Detailed Description . . . . .	80

6.22.2	Constructor & Destructor Documentation	80
6.22.3	Member Function Documentation	80
6.23	mknix::ConstraintThermalFixed Class Reference	80
6.23.1	Detailed Description	82
6.23.2	Constructor & Destructor Documentation	82
6.23.3	Member Function Documentation	82
6.23.4	Member Data Documentation	83
6.24	mknix::Contact Class Reference	83
6.24.1	Detailed Description	83
6.24.2	Constructor & Destructor Documentation	84
6.24.3	Member Function Documentation	84
6.25	Imx::DenseMatrix< T > Class Template Reference	84
6.25.1	Detailed Description	84
6.26	mknix::ElemTetrahedron Class Reference	84
6.26.1	Detailed Description	86
6.26.2	Constructor & Destructor Documentation	86
6.26.3	Member Function Documentation	86
6.27	mknix::ElemTriangle Class Reference	87
6.27.1	Detailed Description	88
6.27.2	Constructor & Destructor Documentation	88
6.27.3	Member Function Documentation	89
6.28	mknix::FlexBody Class Reference	90
6.28.1	Detailed Description	91
6.28.2	Constructor & Destructor Documentation	91
6.28.3	Member Function Documentation	91
6.28.4	Member Data Documentation	95
6.29	mknix::FlexFrameGalerkin Class Reference	96
6.29.1	Detailed Description	97
6.29.2	Constructor & Destructor Documentation	98
6.29.3	Member Function Documentation	98
6.30	mknix::FlexGlobalGalerkin Class Reference	101
6.30.1	Detailed Description	103
6.30.2	Constructor & Destructor Documentation	103
6.30.3	Member Function Documentation	103
6.31	mknix::Force Class Reference	106
6.31.1	Detailed Description	107
6.31.2	Constructor & Destructor Documentation	107
6.31.3	Member Function Documentation	107
6.32	mknix::GaussPoint Class Reference	107
6.32.1	Detailed Description	109

6.32.2	Constructor & Destructor Documentation	109
6.32.3	Member Function Documentation	110
6.32.4	Member Data Documentation	113
6.33	mknix::GaussPoint2D Class Reference	114
6.33.1	Detailed Description	115
6.33.2	Constructor & Destructor Documentation	116
6.33.3	Member Function Documentation	116
6.34	mknix::GaussPoint3D Class Reference	122
6.34.1	Detailed Description	124
6.34.2	Constructor & Destructor Documentation	124
6.34.3	Member Function Documentation	124
6.35	mknix::GaussPointBoundary Class Reference	130
6.35.1	Detailed Description	132
6.35.2	Constructor & Destructor Documentation	132
6.35.3	Member Function Documentation	132
6.35.4	Member Data Documentation	133
6.36	mknix::Load Class Reference	133
6.36.1	Detailed Description	134
6.36.2	Constructor & Destructor Documentation	135
6.36.3	Member Function Documentation	135
6.36.4	Member Data Documentation	135
6.37	mknix::LoadThermal Class Reference	135
6.37.1	Detailed Description	136
6.37.2	Constructor & Destructor Documentation	136
6.37.3	Member Function Documentation	136
6.37.4	Member Data Documentation	137
6.38	mknix::LoadThermalBody Class Reference	137
6.38.1	Detailed Description	137
6.38.2	Constructor & Destructor Documentation	137
6.38.3	Member Function Documentation	138
6.38.4	Member Data Documentation	138
6.39	mknix::LoadThermalBoundary1D Class Reference	138
6.39.1	Detailed Description	139
6.39.2	Constructor & Destructor Documentation	139
6.39.3	Member Function Documentation	139
6.39.4	Member Data Documentation	140
6.40	mknix::Material Class Reference	140
6.40.1	Detailed Description	141
6.40.2	Constructor & Destructor Documentation	141
6.40.3	Member Function Documentation	141

6.41	<a href="#">Imx::Matrix&lt; T &gt; Class Template Reference</a>	146
6.41.1	<a href="#">Detailed Description</a>	146
6.42	<a href="#">mnix::Motion Class Reference</a>	147
6.42.1	<a href="#">Detailed Description</a>	147
6.42.2	<a href="#">Constructor &amp; Destructor Documentation</a>	148
6.42.3	<a href="#">Member Function Documentation</a>	148
6.43	<a href="#">mnix::Node Class Reference</a>	149
6.43.1	<a href="#">Detailed Description</a>	150
6.43.2	<a href="#">Constructor &amp; Destructor Documentation</a>	150
6.43.3	<a href="#">Member Function Documentation</a>	150
6.44	<a href="#">mnix::Point Class Reference</a>	156
6.44.1	<a href="#">Detailed Description</a>	158
6.44.2	<a href="#">Constructor &amp; Destructor Documentation</a>	158
6.44.3	<a href="#">Member Function Documentation</a>	159
6.44.4	<a href="#">Friends And Related Function Documentation</a>	166
6.44.5	<a href="#">Member Data Documentation</a>	167
6.45	<a href="#">mnix::Radiation Class Reference</a>	168
6.45.1	<a href="#">Detailed Description</a>	168
6.45.2	<a href="#">Constructor &amp; Destructor Documentation</a>	169
6.45.3	<a href="#">Member Function Documentation</a>	169
6.46	<a href="#">mnix::Reader Class Reference</a>	169
6.46.1	<a href="#">Detailed Description</a>	169
6.46.2	<a href="#">Constructor &amp; Destructor Documentation</a>	169
6.46.3	<a href="#">Member Function Documentation</a>	170
6.47	<a href="#">mnix::ReaderConstraints Class Reference</a>	170
6.47.1	<a href="#">Detailed Description</a>	170
6.47.2	<a href="#">Constructor &amp; Destructor Documentation</a>	171
6.47.3	<a href="#">Member Function Documentation</a>	171
6.48	<a href="#">mnix::ReaderFlex Class Reference</a>	171
6.48.1	<a href="#">Detailed Description</a>	171
6.48.2	<a href="#">Constructor &amp; Destructor Documentation</a>	172
6.48.3	<a href="#">Member Function Documentation</a>	172
6.49	<a href="#">mnix::ReaderRigid Class Reference</a>	172
6.49.1	<a href="#">Detailed Description</a>	173
6.49.2	<a href="#">Constructor &amp; Destructor Documentation</a>	173
6.49.3	<a href="#">Member Function Documentation</a>	173
6.50	<a href="#">mnix::RigidBar Class Reference</a>	173
6.50.1	<a href="#">Detailed Description</a>	175
6.50.2	<a href="#">Constructor &amp; Destructor Documentation</a>	175
6.50.3	<a href="#">Member Function Documentation</a>	175

6.51	<a href="#">mknix::RigidBody Class Reference</a>	177
6.51.1	<a href="#">Detailed Description</a>	179
6.51.2	<a href="#">Constructor &amp; Destructor Documentation</a>	179
6.51.3	<a href="#">Member Function Documentation</a>	179
6.51.4	<a href="#">Member Data Documentation</a>	182
6.52	<a href="#">mknix::RigidBody2D Class Reference</a>	182
6.52.1	<a href="#">Detailed Description</a>	184
6.52.2	<a href="#">Constructor &amp; Destructor Documentation</a>	184
6.52.3	<a href="#">Member Function Documentation</a>	184
6.53	<a href="#">mknix::RigidBody3D Class Reference</a>	186
6.53.1	<a href="#">Detailed Description</a>	187
6.53.2	<a href="#">Constructor &amp; Destructor Documentation</a>	187
6.53.3	<a href="#">Member Function Documentation</a>	187
6.54	<a href="#">mknix::RigidBodyMassPoint Class Reference</a>	188
6.54.1	<a href="#">Detailed Description</a>	190
6.54.2	<a href="#">Constructor &amp; Destructor Documentation</a>	190
6.54.3	<a href="#">Member Function Documentation</a>	190
6.55	<a href="#">mknix::ShapeFunction Class Reference</a>	191
6.55.1	<a href="#">Detailed Description</a>	192
6.55.2	<a href="#">Constructor &amp; Destructor Documentation</a>	192
6.55.3	<a href="#">Member Function Documentation</a>	193
6.55.4	<a href="#">Member Data Documentation</a>	195
6.56	<a href="#">mknix::ShapeFunctionLinear Class Reference</a>	195
6.56.1	<a href="#">Detailed Description</a>	196
6.56.2	<a href="#">Constructor &amp; Destructor Documentation</a>	197
6.56.3	<a href="#">Member Function Documentation</a>	197
6.57	<a href="#">mknix::ShapeFunctionLinearX Class Reference</a>	197
6.57.1	<a href="#">Detailed Description</a>	198
6.57.2	<a href="#">Constructor &amp; Destructor Documentation</a>	199
6.57.3	<a href="#">Member Function Documentation</a>	199
6.58	<a href="#">mknix::ShapeFunctionMLS Class Reference</a>	199
6.58.1	<a href="#">Detailed Description</a>	201
6.58.2	<a href="#">Constructor &amp; Destructor Documentation</a>	201
6.58.3	<a href="#">Member Function Documentation</a>	201
6.59	<a href="#">mknix::ShapeFunctionRBF Class Reference</a>	202
6.59.1	<a href="#">Detailed Description</a>	203
6.59.2	<a href="#">Constructor &amp; Destructor Documentation</a>	203
6.59.3	<a href="#">Member Function Documentation</a>	203
6.60	<a href="#">mknix::ShapeFunctionTetrahedron Class Reference</a>	203
6.60.1	<a href="#">Detailed Description</a>	204



6.60.2	Constructor & Destructor Documentation	205
6.60.3	Member Function Documentation	205
6.61	mknix::ShapeFunctionTriangle Class Reference	205
6.61.1	Detailed Description	206
6.61.2	Constructor & Destructor Documentation	207
6.61.3	Member Function Documentation	207
6.62	mknix::ShapeFunctionTriangleSigned Class Reference	207
6.62.1	Detailed Description	208
6.62.2	Constructor & Destructor Documentation	208
6.62.3	Member Function Documentation	209
6.63	mknix::Simulation Class Reference	209
6.63.1	Detailed Description	210
6.63.2	Constructor & Destructor Documentation	211
6.63.3	Member Function Documentation	211
6.63.4	Friends And Related Function Documentation	227
6.64	mknix::System Class Reference	228
6.64.1	Detailed Description	229
6.64.2	Constructor & Destructor Documentation	230
6.64.3	Member Function Documentation	230
6.64.4	Friends And Related Function Documentation	233
6.64.5	Member Data Documentation	233
6.65	mknix::SystemChain Class Reference	234
6.65.1	Detailed Description	235
6.65.2	Constructor & Destructor Documentation	236
6.65.3	Member Function Documentation	236
6.66	mknix::ThermalBody Class Reference	237
6.66.1	Detailed Description	239
6.66.2	Constructor & Destructor Documentation	239
6.66.3	Member Function Documentation	239
6.66.4	Member Data Documentation	242
6.67	Imx::Vector< T > Class Template Reference	242
6.67.1	Detailed Description	243
<b>7</b>	<b>File Documentation</b>	<b>243</b>
7.1	analysis.cpp File Reference	243
7.2	analysis.h File Reference	243
7.3	analysisdynamic.cpp File Reference	244
7.4	analysisdynamic.h File Reference	245
7.5	analysisstatic.cpp File Reference	246
7.6	analysisstatic.h File Reference	247

7.7	<a href="#">analysisthermaldynamic.cpp File Reference</a>	247
7.8	<a href="#">analysisthermaldynamic.h File Reference</a>	248
7.9	<a href="#">analysisthermalstatic.cpp File Reference</a>	249
7.10	<a href="#">analysisthermalstatic.h File Reference</a>	250
7.11	<a href="#">analysisthermomechanicaldynamic.cpp File Reference</a>	250
7.12	<a href="#">analysisthermomechanicaldynamic.h File Reference</a>	251
7.13	<a href="#">body.cpp File Reference</a>	252
7.14	<a href="#">body.h File Reference</a>	253
7.15	<a href="#">bodyflex.cpp File Reference</a>	254
7.16	<a href="#">bodyflex.h File Reference</a>	254
7.17	<a href="#">bodyflexframegalerkin.cpp File Reference</a>	255
7.18	<a href="#">bodyflexframegalerkin.h File Reference</a>	256
7.19	<a href="#">bodyflexglobalgalerkin.cpp File Reference</a>	258
7.20	<a href="#">bodyflexglobalgalerkin.h File Reference</a>	258
7.21	<a href="#">bodyrigid.cpp File Reference</a>	260
7.22	<a href="#">bodyrigid.h File Reference</a>	260
7.23	<a href="#">bodyrigid0D.cpp File Reference</a>	261
7.24	<a href="#">bodyrigid0D.h File Reference</a>	262
7.25	<a href="#">bodyrigid1D.cpp File Reference</a>	264
7.26	<a href="#">bodyrigid1D.h File Reference</a>	264
7.27	<a href="#">bodyrigid2D.cpp File Reference</a>	266
7.28	<a href="#">bodyrigid2D.h File Reference</a>	266
7.29	<a href="#">bodyrigid3D.cpp File Reference</a>	268
7.30	<a href="#">bodyrigid3D.h File Reference</a>	268
7.31	<a href="#">bodythermal.cpp File Reference</a>	270
7.32	<a href="#">bodythermal.h File Reference</a>	270
7.33	<a href="#">boundarygroup.cpp File Reference</a>	271
7.34	<a href="#">boundarygroup.h File Reference</a>	272
7.35	<a href="#">cell.cpp File Reference</a>	273
7.36	<a href="#">cell.h File Reference</a>	274
7.36.1	Detailed Description	274
7.37	<a href="#">cellboundary.cpp File Reference</a>	275
7.38	<a href="#">cellboundary.h File Reference</a>	275
7.39	<a href="#">cellboundarylinear.cpp File Reference</a>	276
7.40	<a href="#">cellboundarylinear.h File Reference</a>	277
7.40.1	Detailed Description	278
7.41	<a href="#">cellrect.cpp File Reference</a>	278
7.42	<a href="#">cellrect.h File Reference</a>	279
7.42.1	Detailed Description	280
7.43	<a href="#">celltetrahedron.cpp File Reference</a>	281

7.44	celltetrahedron.h File Reference	281
7.44.1	Detailed Description	283
7.45	celltriang.cpp File Reference	283
7.46	celltriang.h File Reference	283
7.46.1	Detailed Description	285
7.47	common.cpp File Reference	285
7.48	common.h File Reference	285
7.49	compbar.cpp File Reference	286
7.50	compbar.h File Reference	286
7.51	constraint.cpp File Reference	287
7.52	constraint.h File Reference	287
7.53	constraintclearance.cpp File Reference	288
7.54	constraintclearance.h File Reference	289
7.55	constraintcontact.cpp File Reference	290
7.56	constraintcontact.h File Reference	291
7.57	constraintdistance.cpp File Reference	292
7.58	constraintdistance.h File Reference	292
7.59	constraintfixedaxis.cpp File Reference	293
7.60	constraintfixedaxis.h File Reference	294
7.61	constraintfixedcoordinates.cpp File Reference	295
7.62	constraintfixedcoordinates.h File Reference	296
7.63	constraintthermal.cpp File Reference	297
7.64	constraintthermal.h File Reference	297
7.65	constraintthermalfixed.cpp File Reference	298
7.66	constraintthermalfixed.h File Reference	299
7.67	dictionary.h File Reference	301
7.68	elemtetrahedron.cpp File Reference	301
7.69	elemtetrahedron.h File Reference	301
7.70	elemtriangle.cpp File Reference	303
7.71	elemtriangle.h File Reference	303
7.72	force.cpp File Reference	305
7.73	force.h File Reference	305
7.74	gausspoint.cpp File Reference	306
7.75	gausspoint.h File Reference	307
7.75.1	Detailed Description	308
7.76	gausspoint2D.cpp File Reference	308
7.77	gausspoint2D.h File Reference	309
7.78	gausspoint3D.cpp File Reference	310
7.79	gausspoint3D.h File Reference	310
7.80	gausspointboundary.cpp File Reference	312

7.81	<a href="#">gausspointboundary.h File Reference</a>	312
7.82	<a href="#">generalcontact.cpp File Reference</a>	313
7.83	<a href="#">generalcontact.h File Reference</a>	313
7.84	<a href="#">load.cpp File Reference</a>	314
7.85	<a href="#">load.h File Reference</a>	315
7.86	<a href="#">loadradiation.cpp File Reference</a>	316
7.87	<a href="#">loadradiation.h File Reference</a>	316
7.88	<a href="#">loadthermal.cpp File Reference</a>	317
7.89	<a href="#">loadthermal.h File Reference</a>	318
7.90	<a href="#">loadthermalbody.cpp File Reference</a>	319
7.91	<a href="#">loadthermalbody.h File Reference</a>	320
7.92	<a href="#">loadthermalboundary1D.cpp File Reference</a>	320
7.93	<a href="#">loadthermalboundary1D.h File Reference</a>	321
7.94	<a href="#">material.cpp File Reference</a>	322
7.95	<a href="#">material.h File Reference</a>	323
7.96	<a href="#">motion.cpp File Reference</a>	323
7.97	<a href="#">motion.h File Reference</a>	324
7.98	<a href="#">node.cpp File Reference</a>	325
7.99	<a href="#">node.h File Reference</a>	326
7.100	<a href="#">point.cpp File Reference</a>	327
7.101	<a href="#">point.h File Reference</a>	328
7.101.1	Detailed Description	329
7.102	<a href="#">reader.cpp File Reference</a>	329
7.103	<a href="#">reader.h File Reference</a>	329
7.104	<a href="#">readerconstraints.cpp File Reference</a>	330
7.105	<a href="#">readerconstraints.h File Reference</a>	331
7.106	<a href="#">readerflex.cpp File Reference</a>	332
7.107	<a href="#">readerflex.h File Reference</a>	333
7.108	<a href="#">readerrigid.cpp File Reference</a>	333
7.109	<a href="#">readerrigid.h File Reference</a>	334
7.110	<a href="#">shapefunction.cpp File Reference</a>	335
7.111	<a href="#">shapefunction.h File Reference</a>	336
7.111.1	Detailed Description	336
7.112	<a href="#">shapefunctionlinear-x.cpp File Reference</a>	336
7.113	<a href="#">shapefunctionlinear-x.h File Reference</a>	337
7.114	<a href="#">shapefunctionlinear.cpp File Reference</a>	338
7.115	<a href="#">shapefunctionlinear.h File Reference</a>	339
7.116	<a href="#">shapefunctionMLS.cpp File Reference</a>	340
7.117	<a href="#">shapefunctionMLS.h File Reference</a>	340
7.117.1	Detailed Description	341

7.118shapefunctionMLS2D.cpp File Reference . . . . .	342
7.119shapefunctionMLS2D.h File Reference . . . . .	343
7.120shapefunctionMLS3D.cpp File Reference . . . . .	343
7.121shapefunctionMLS3D.h File Reference . . . . .	344
7.122shapefunctionRBF.cpp File Reference . . . . .	345
7.123shapefunctionRBF.h File Reference . . . . .	346
7.123.1 Detailed Description . . . . .	346
7.124shapefunctiontetrahedron.cpp File Reference . . . . .	347
7.125shapefunctiontetrahedron.h File Reference . . . . .	348
7.126shapefunctiontriangle.cpp File Reference . . . . .	348
7.127shapefunctiontriangle.h File Reference . . . . .	349
7.128shapefunctiontriangle3D.cpp File Reference . . . . .	350
7.129shapefunctiontriangle3D.h File Reference . . . . .	351
7.130simulation.cpp File Reference . . . . .	352
7.131simulation.h File Reference . . . . .	352
7.132system.cpp File Reference . . . . .	353
7.133system.h File Reference . . . . .	354
7.134systemchain.cpp File Reference . . . . .	355
7.135systemchain.h File Reference . . . . .	355
7.136systemchain2.cpp File Reference . . . . .	356
7.137systemchain2.h File Reference . . . . .	357
<b>Index</b>	<b>359</b>

## 1 Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<b>Imx</b>	<b>??</b>
<b>mknix</b>	<b>??</b>

## 2 Hierarchical Index

### 2.1 Class Hierarchy

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

<b>mknix::Analysis</b>	<b>13</b>
<b>mknix::AnalysisDynamic</b>	<b>17</b>
<b>mknix::AnalysisStatic</b>	<b>19</b>

mknix::AnalysisThermalDynamic	21
mknix::AnalysisThermalStatic	24
mknix::AnalysisThermoMechanicalDynamic	25
mknix::Body	28
mknix::FlexBody	90
mknix::FlexFrameGalerkin	96
mknix::FlexGlobalGalerkin	101
mknix::RigidBody	177
mknix::RigidBar	173
mknix::RigidBody2D	182
mknix::RigidBody3D	186
mknix::RigidBodyMassPoint	188
mknix::BoundaryGroup	38
mknix::Cell	40
mknix::CellRect	51
mknix::CellTetrahedron	53
mknix::ElemTetrahedron	84
mknix::CellTriang	55
mknix::ElemTriangle	87
mknix::CellBoundary	45
mknix::CellBoundaryLinear	47
mknix::CompBar	59
mknix::Constraint	59
mknix::ConstraintClearance	65
mknix::ConstraintContact	67
mknix::ConstraintDistance	70
mknix::ConstraintFixedAxis	73
mknix::ConstraintFixedCoordinates	76
mknix::ConstraintThermal	78
mknix::ConstraintThermalFixed	80
mknix::Contact	83
Imx::DenseMatrix< T >	84

<b>Imx::DenseMatrix&lt; data_type &gt;</b>	<b>84</b>
<b>Imx::DenseMatrix&lt; double &gt;</b>	<b>84</b>
<b>mnix::Load</b>	<b>133</b>
mnix::Force	106
mnix::Radiation	168
<b>mnix::LoadThermal</b>	<b>135</b>
<b>mnix::LoadThermalBody</b>	<b>137</b>
<b>mnix::LoadThermalBoundary1D</b>	<b>138</b>
<b>mnix::Material</b>	<b>140</b>
<b>Imx::Matrix&lt; T &gt;</b>	<b>146</b>
<b>Imx::Matrix&lt; data_type &gt;</b>	<b>146</b>
<b>mnix::Point</b>	<b>156</b>
mnix::GaussPoint	107
mnix::GaussPoint2D	114
mnix::GaussPoint3D	122
mnix::GaussPointBoundary	130
mnix::Node	149
<b>mnix::Reader</b>	<b>169</b>
<b>mnix::ReaderConstraints</b>	<b>170</b>
<b>mnix::ReaderFlex</b>	<b>171</b>
<b>mnix::ReaderRigid</b>	<b>172</b>
<b>mnix::ShapeFunction</b>	<b>191</b>
mnix::ShapeFunctionLinear	195
mnix::ShapeFunctionLinearX	197
mnix::ShapeFunctionMLS	199
mnix::ShapeFunctionMLS	199
mnix::ShapeFunctionMLS	199
mnix::ShapeFunctionRBF	202
mnix::ShapeFunctionTetrahedron	203
mnix::ShapeFunctionTriangle	205
mnix::ShapeFunctionTriangleSigned	207
<b>mnix::Simulation</b>	<b>209</b>

<b>mknix::System</b>	<b>228</b>
<b>mknix::Motion</b>	<b>147</b>
<b>mknix::SystemChain</b>	<b>234</b>
<b>mknix::SystemChain</b>	<b>234</b>
<b>mknix::ThermalBody</b>	<b>237</b>
<b>lmx::Vector&lt; T &gt;</b>	<b>242</b>
<b>lmx::Vector&lt; data_type &gt;</b>	<b>242</b>
<b>lmx::Vector&lt; double &gt;</b>	<b>242</b>

### 3 Class Index

#### 3.1 Class List

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

<b>mknix::Analysis</b>	<b>??</b>
<b>mknix::AnalysisDynamic</b>	<b>??</b>
<b>mknix::AnalysisStatic</b>	<b>??</b>
<b>mknix::AnalysisThermalDynamic</b>	<b>??</b>
<b>mknix::AnalysisThermalStatic</b>	<b>??</b>
<b>mknix::AnalysisThermoMechanicalDynamic</b>	<b>??</b>
<b>mknix::Body</b>	<b>??</b>
<b>mknix::BoundaryGroup</b>	<b>??</b>
<b>mknix::Cell</b>	<b>??</b>
<b>mknix::CellBoundary</b>	<b>??</b>
<b>mknix::CellBoundaryLinear</b>	<b>??</b>
<b>mknix::CellRect</b>	<b>??</b>
<b>mknix::CellTetrahedron</b>	<b>??</b>
<b>mknix::CellTriang</b>	<b>??</b>
<b>mknix::CompBar</b>	<b>??</b>
<b>mknix::Constraint</b>	<b>??</b>
<b>mknix::ConstraintClearance</b>	<b>??</b>
<b>mknix::ConstraintContact</b>	<b>??</b>
<b>mknix::ConstraintDistance</b>	<b>??</b>
<b>mknix::ConstraintFixedAxis</b>	<b>??</b>



<a href="#">mknix::ConstraintFixedCoordinates</a>	??
<a href="#">mknix::ConstraintThermal</a>	??
<a href="#">mknix::ConstraintThermalFixed</a>	??
<a href="#">mknix::Contact</a>	??
<a href="#">Imx::DenseMatrix&lt; T &gt;</a>	??
<a href="#">mknix::ElemTetrahedron</a>	??
<a href="#">mknix::ElemTriangle</a>	??
<a href="#">mknix::FlexBody</a>	??
<a href="#">mknix::FlexFrameGalerkin</a>	??
<a href="#">mknix::FlexGlobalGalerkin</a>	??
<a href="#">mknix::Force</a>	??
<a href="#">mknix::GaussPoint</a>	??
<a href="#">mknix::GaussPoint2D</a>	??
<a href="#">mknix::GaussPoint3D</a>	??
<a href="#">mknix::GaussPointBoundary</a>	??
<a href="#">mknix::Load</a>	??
<a href="#">mknix::LoadThermal</a>	??
<a href="#">mknix::LoadThermalBody</a>	??
<a href="#">mknix::LoadThermalBoundary1D</a>	??
<a href="#">mknix::Material</a>	??
<a href="#">Imx::Matrix&lt; T &gt;</a>	??
<a href="#">mknix::Motion</a>	??
<a href="#">mknix::Node</a>	??
<a href="#">mknix::Point</a>	??
<a href="#">mknix::Radiation</a>	??
<a href="#">mknix::Reader</a>	??
<a href="#">mknix::ReaderConstraints</a>	??
<a href="#">mknix::ReaderFlex</a>	??
<a href="#">mknix::ReaderRigid</a>	??
<a href="#">mknix::RigidBar</a>	??
<a href="#">mknix::RigidBody</a>	??
<a href="#">mknix::RigidBody2D</a>	??

<a href="#">mknix::RigidBody3D</a>	??
<a href="#">mknix::RigidBodyMassPoint</a>	??
<a href="#">mknix::ShapeFunction</a>	??
<a href="#">mknix::ShapeFunctionLinear</a>	??
<a href="#">mknix::ShapeFunctionLinearX</a>	??
<a href="#">mknix::ShapeFunctionMLS</a>	??
<a href="#">mknix::ShapeFunctionRBF</a>	??
<a href="#">mknix::ShapeFunctionTetrahedron</a>	??
<a href="#">mknix::ShapeFunctionTriangle</a>	??
<a href="#">mknix::ShapeFunctionTriangleSigned</a>	??
<a href="#">mknix::Simulation</a>	??
<a href="#">mknix::System</a>	??
<a href="#">mknix::SystemChain</a>	??
<a href="#">mknix::ThermalBody</a>	??
<a href="#">lmx::Vector&lt; T &gt;</a>	??

## 4 File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">analysis.cpp</a>	??
<a href="#">analysis.h</a>	??
<a href="#">analysisdynamic.cpp</a>	??
<a href="#">analysisdynamic.h</a>	??
<a href="#">analysisstatic.cpp</a>	??
<a href="#">analysisstatic.h</a>	??
<a href="#">analysisthermaldynamic.cpp</a>	??
<a href="#">analysisthermaldynamic.h</a>	??
<a href="#">analysisthermalstatic.cpp</a>	??
<a href="#">analysisthermalstatic.h</a>	??
<a href="#">analysisthermomechanicaldynamic.cpp</a>	??
<a href="#">analysisthermomechanicaldynamic.h</a>	??
<a href="#">body.cpp</a>	??

<a href="#">body.h</a>	??
<a href="#">bodyflex.cpp</a>	??
<a href="#">bodyflex.h</a>	??
<a href="#">bodyflexframegalerkin.cpp</a>	??
<a href="#">bodyflexframegalerkin.h</a>	??
<a href="#">bodyflexglobalgalerkin.cpp</a>	??
<a href="#">bodyflexglobalgalerkin.h</a>	??
<a href="#">bodyrigid.cpp</a>	??
<a href="#">bodyrigid.h</a>	??
<a href="#">bodyrigid0D.cpp</a>	??
<a href="#">bodyrigid0D.h</a>	??
<a href="#">bodyrigid1D.cpp</a>	??
<a href="#">bodyrigid1D.h</a>	??
<a href="#">bodyrigid2D.cpp</a>	??
<a href="#">bodyrigid2D.h</a>	??
<a href="#">bodyrigid3D.cpp</a>	??
<a href="#">bodyrigid3D.h</a>	??
<a href="#">bodythermal.cpp</a>	??
<a href="#">bodythermal.h</a>	??
<a href="#">boundarygroup.cpp</a>	??
<a href="#">boundarygroup.h</a>	??
<a href="#">cell.cpp</a>	??
<a href="#">cell.h</a>	
Background cells for integration	??
<a href="#">cellboundary.cpp</a>	??
<a href="#">cellboundary.h</a>	??
<a href="#">cellboundarylinear.cpp</a>	??
<a href="#">cellboundarylinear.h</a>	
Background cells for boundary integration	??
<a href="#">cellrect.cpp</a>	??
<a href="#">cellrect.h</a>	
Background cells for integration	??
<a href="#">celltetrahedron.cpp</a>	??

<a href="#">celltetrahedron.h</a>	
Background cells for integration	??
<a href="#">celltriang.cpp</a>	??
<a href="#">celltriang.h</a>	
Background cells for integration	??
<a href="#">common.cpp</a>	??
<a href="#">common.h</a>	??
<a href="#">compbar.cpp</a>	??
<a href="#">compbar.h</a>	??
<a href="#">constraint.cpp</a>	??
<a href="#">constraint.h</a>	??
<a href="#">constraintclearance.cpp</a>	??
<a href="#">constraintclearance.h</a>	??
<a href="#">constraintcontact.cpp</a>	??
<a href="#">constraintcontact.h</a>	??
<a href="#">constraintdistance.cpp</a>	??
<a href="#">constraintdistance.h</a>	??
<a href="#">constraintfixedaxis.cpp</a>	??
<a href="#">constraintfixedaxis.h</a>	??
<a href="#">constraintfixedcoordinates.cpp</a>	??
<a href="#">constraintfixedcoordinates.h</a>	??
<a href="#">constraintthermal.cpp</a>	??
<a href="#">constraintthermal.h</a>	??
<a href="#">constraintthermalfixed.cpp</a>	??
<a href="#">constraintthermalfixed.h</a>	??
<a href="#">dictionary.h</a>	??
<a href="#">elemtetrahedron.cpp</a>	??
<a href="#">elemtetrahedron.h</a>	??
<a href="#">elemtriangle.cpp</a>	??
<a href="#">elemtriangle.h</a>	??
<a href="#">force.cpp</a>	??
<a href="#">force.h</a>	??
<a href="#">gausspoint.cpp</a>	??

<a href="#">gausspoint.h</a>	
Point for numerical integration	??
<a href="#">gausspoint2D.cpp</a>	??
<a href="#">gausspoint2D.h</a>	??
<a href="#">gausspoint3D.cpp</a>	??
<a href="#">gausspoint3D.h</a>	??
<a href="#">gausspointboundary.cpp</a>	??
<a href="#">gausspointboundary.h</a>	??
<a href="#">generalcontact.cpp</a>	??
<a href="#">generalcontact.h</a>	??
<a href="#">load.cpp</a>	??
<a href="#">load.h</a>	??
<a href="#">loadradiation.cpp</a>	??
<a href="#">loadradiation.h</a>	??
<a href="#">loadthermal.cpp</a>	??
<a href="#">loadthermal.h</a>	??
<a href="#">loadthermalbody.cpp</a>	??
<a href="#">loadthermalbody.h</a>	??
<a href="#">loadthermalboundary1D.cpp</a>	??
<a href="#">loadthermalboundary1D.h</a>	??
<a href="#">material.cpp</a>	??
<a href="#">material.h</a>	??
<a href="#">motion.cpp</a>	??
<a href="#">motion.h</a>	??
<a href="#">node.cpp</a>	??
<a href="#">node.h</a>	??
<a href="#">point.cpp</a>	??
<a href="#">point.h</a>	
Point of interest	??
<a href="#">reader.cpp</a>	??
<a href="#">reader.h</a>	??
<a href="#">readerconstraints.cpp</a>	??
<a href="#">readerconstraints.h</a>	??

<a href="#">readerflex.cpp</a>	??
<a href="#">readerflex.h</a>	??
<a href="#">readerrigid.cpp</a>	??
<a href="#">readerrigid.h</a>	??
<a href="#">shapefunction.cpp</a>	??
<a href="#">shapefunction.h</a>	
Function for interpolation of basic variables	??
<a href="#">shapefunctionlinear-x.cpp</a>	??
<a href="#">shapefunctionlinear-x.h</a>	??
<a href="#">shapefunctionlinear.cpp</a>	??
<a href="#">shapefunctionlinear.h</a>	??
<a href="#">shapefunctionMLS.cpp</a>	??
<a href="#">shapefunctionMLS.h</a>	
Function for aproximation of basic variables by Moving Least Squares fit	??
<a href="#">shapefunctionMLS2D.cpp</a>	??
<a href="#">shapefunctionMLS2D.h</a>	??
<a href="#">shapefunctionMLS3D.cpp</a>	??
<a href="#">shapefunctionMLS3D.h</a>	??
<a href="#">shapefunctionRBF.cpp</a>	??
<a href="#">shapefunctionRBF.h</a>	
Function for interpolation of basic variables by means of Radial basis Functions	??
<a href="#">shapefunctiontetrahedron.cpp</a>	??
<a href="#">shapefunctiontetrahedron.h</a>	??
<a href="#">shapefunctiontriangle.cpp</a>	??
<a href="#">shapefunctiontriangle.h</a>	??
<a href="#">shapefunctiontriangle3D.cpp</a>	??
<a href="#">shapefunctiontriangle3D.h</a>	??
<a href="#">simulation.cpp</a>	??
<a href="#">simulation.h</a>	??
<a href="#">system.cpp</a>	??
<a href="#">system.h</a>	??
<a href="#">systemchain.cpp</a>	??
<a href="#">systemchain.h</a>	??

<a href="#">systemchain2.cpp</a>	??
<a href="#">systemchain2.h</a>	??

## 5 Namespace Documentation

### 5.1 Imx Namespace Reference

#### Classes

- class [DenseMatrix](#)
- class [Matrix](#)
- class [Vector](#)

### 5.2 mknix Namespace Reference

#### Classes

- class [Analysis](#)
- class [AnalysisDynamic](#)
- class [AnalysisStatic](#)
- class [AnalysisThermalDynamic](#)
- class [AnalysisThermalStatic](#)
- class [AnalysisThermoMechanicalDynamic](#)
- class [Body](#)
- class [BoundaryGroup](#)
- class [Cell](#)
- class [CellBoundary](#)
- class [CellBoundaryLinear](#)
- class [CellRect](#)
- class [CellTetrahedron](#)
- class [CellTriang](#)
- class [CompBar](#)
- class [Constraint](#)
- class [ConstraintClearance](#)
- class [ConstraintContact](#)
- class [ConstraintDistance](#)
- class [ConstraintFixedAxis](#)
- class [ConstraintFixedCoordinates](#)
- class [ConstraintThermal](#)
- class [ConstraintThermalFixed](#)
- class [Contact](#)
- class [ElemTetrahedron](#)
- class [ElemTriangle](#)
- class [FlexBody](#)
- class [FlexFrameGalerkin](#)
- class [FlexGlobalGalerkin](#)
- class [Force](#)
- class [GaussPoint](#)
- class [GaussPoint2D](#)
- class [GaussPoint3D](#)
- class [GaussPointBoundary](#)
- class [Load](#)

- class [LoadThermal](#)
- class [LoadThermalBody](#)
- class [LoadThermalBoundary1D](#)
- class [Material](#)
- class [Motion](#)
- class [Node](#)
- class [Point](#)
- class [Radiation](#)
- class [Reader](#)
- class [ReaderConstraints](#)
- class [ReaderFlex](#)
- class [ReaderRigid](#)
- class [RigidBar](#)
- class [RigidBody](#)
- class [RigidBody2D](#)
- class [RigidBody3D](#)
- class [RigidBodyMassPoint](#)
- class [ShapeFunction](#)
- class [ShapeFunctionLinear](#)
- class [ShapeFunctionLinearX](#)
- class [ShapeFunctionMLS](#)
- class [ShapeFunctionRBF](#)
- class [ShapeFunctionTetrahedron](#)
- class [ShapeFunctionTriangle](#)
- class [ShapeFunctionTriangleSigned](#)
- class [Simulation](#)
- class [System](#)
- class [SystemChain](#)
- class [ThermalBody](#)

## Typedefs

- typedef double [data\\_type](#)

## Functions

- double [interpolate1D](#) (double key, const std::map< double, double > &theMap)
- template<class T , class... Args>  
std::unique\_ptr< T > [make\\_unique](#) (Args &&...args)
- void [setSignal](#) (std::string node, std::vector< double >)
- std::vector< double > [getSignal](#) (std::string node)

### 5.2.1 Detailed Description

#### Author

Daniel Iglesias <daniel>

### 5.2.2 Typedef Documentation

#### 5.2.2.1 typedef double mknix::data\_type

Definition at line 28 of file common.h.



### 5.2.3 Function Documentation

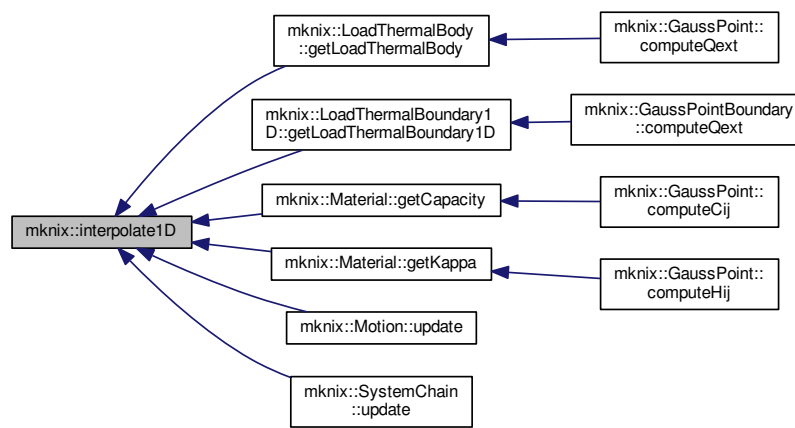
#### 5.2.3.1 `std::vector<double> mknix::getSignal ( std::string node )`

Definition at line 208 of file simulation.cpp.

#### 5.2.3.2 `double mknix::interpolate1D ( double key, const std::map< double, double > &theMap )`

Definition at line 24 of file common.cpp.

Here is the caller graph for this function:



#### 5.2.3.3 `template<class T, class... Args> std::unique_ptr<T> mknix::make_unique ( Args &&... args )`

Stand-in for `std::make_unique` included in C++14

Definition at line 36 of file common.h.

#### 5.2.3.4 `void mknix::setSignal ( std::string node, std::vector< double > )`

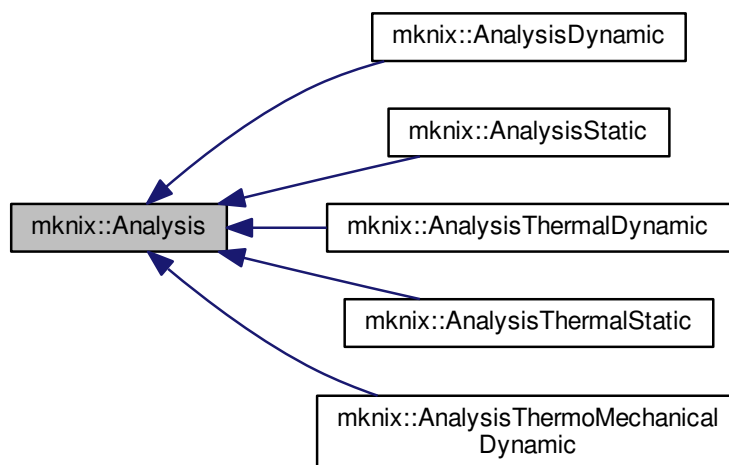
Definition at line 202 of file simulation.cpp.

## 6 Class Documentation

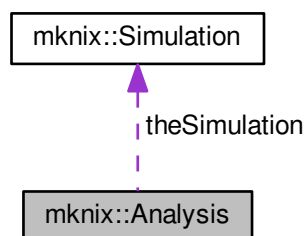
### 6.1 mknix::Analysis Class Reference

```
#include <analysis.h>
```

Inheritance diagram for `mknix::Analysis`:



Collaboration diagram for `mknix::Analysis`:



#### Public Member Functions

- `Analysis ()`
- `Analysis (Simulation *)`
- `virtual ~Analysis ()`
- `virtual std::string type ()=0`
- `void setEpsilon (double epsilon_in)`
- `virtual void solve (Imx::Vector< data_type > *, Imx::Vector< data_type > *=0, Imx::Vector< data_type > *=0)=0`
- `virtual void init (Imx::Vector< data_type > *)`
- `virtual void nextStep ()`

## Protected Attributes

- [Simulation](#) \* [theSimulation](#)
- double [epsilon](#)

## 6.1.1 Detailed Description

## Author

AUTHORS <MAILS>

Definition at line 35 of file analysis.h.

## 6.1.2 Constructor &amp; Destructor Documentation

## 6.1.2.1 mknix::Analysis::Analysis ( )

Definition at line 25 of file analysis.cpp.

6.1.2.2 mknix::Analysis::Analysis ( [Simulation](#) \* *simulation\_in* )

Definition at line 30 of file analysis.cpp.

## 6.1.2.3 mknix::Analysis::~~Analysis ( ) [virtual]

Definition at line 37 of file analysis.cpp.

## 6.1.3 Member Function Documentation

6.1.3.1 virtual void mknix::Analysis::init ( [Imx::Vector](#)< [data\\_type](#) > \* ) [inline],[virtual]

Reimplemented in [mknix::AnalysisThermalDynamic](#).

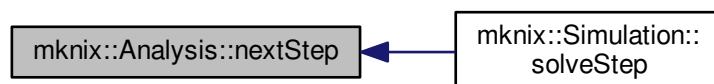
Definition at line 54 of file analysis.h.

## 6.1.3.2 virtual void mknix::Analysis::nextStep ( ) [inline],[virtual]

Reimplemented in [mknix::AnalysisThermalDynamic](#).

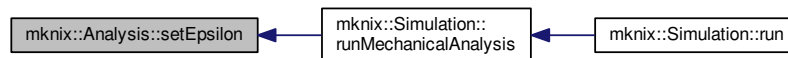
Definition at line 55 of file analysis.h.

Here is the caller graph for this function:

6.1.3.3 void mknix::Analysis::setEpsilon ( double *epsilon\_in* ) [inline]

Definition at line 45 of file analysis.h.

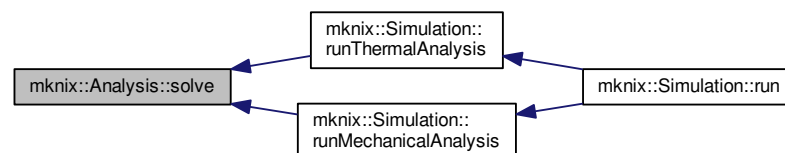
Here is the caller graph for this function:



**6.1.3.4** `virtual void mknix::Analysis::solve ( Imx::Vector< data_type > *, Imx::Vector< data_type > * = 0, Imx::Vector< data_type > * = 0 ) [pure virtual]`

Implemented in [mknix::AnalysisThermalDynamic](#), [mknix::AnalysisThermoMechanicalDynamic](#), [mknix::AnalysisDynamic](#), [mknix::AnalysisStatic](#), and [mknix::AnalysisThermalStatic](#).

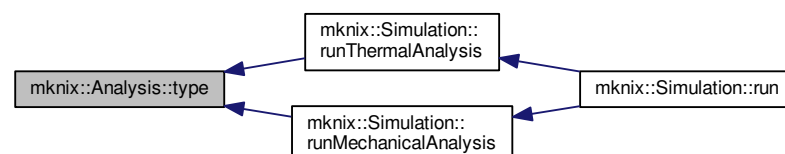
Here is the caller graph for this function:



**6.1.3.5** `virtual std::string mknix::Analysis::type ( ) [pure virtual]`

Implemented in [mknix::AnalysisThermoMechanicalDynamic](#), [mknix::AnalysisDynamic](#), [mknix::AnalysisStatic](#), [mknix::AnalysisThermalDynamic](#), and [mknix::AnalysisThermalStatic](#).

Here is the caller graph for this function:



## 6.1.4 Member Data Documentation

**6.1.4.1** `double mknix::Analysis::epsilon [protected]`

Definition at line 59 of file analysis.h.

**6.1.4.2** `Simulation* mknix::Analysis::theSimulation [protected]`

Definition at line 58 of file analysis.h.

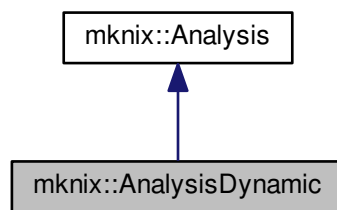
The documentation for this class was generated from the following files:

- [analysis.h](#)
- [analysis.cpp](#)

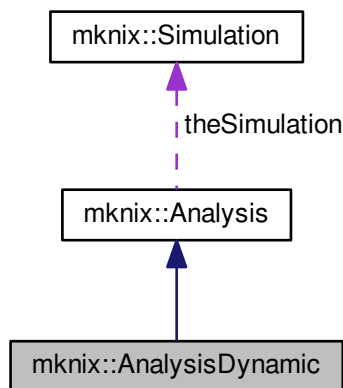
## 6.2 mknix::AnalysisDynamic Class Reference

```
#include <analysisdynamic.h>
```

Inheritance diagram for mknix::AnalysisDynamic:



Collaboration diagram for mknix::AnalysisDynamic:



### Public Member Functions

- [AnalysisDynamic](#) ()
- [AnalysisDynamic](#) ([Simulation](#) \*, double, double, double, char \*, double, double, double)
- [~AnalysisDynamic](#) ()
- `std::string type` ()
- `void solve` ([lmx::Vector](#)< [data\\_type](#) > \*, [lmx::Vector](#)< [data\\_type](#) > \*, [lmx::Vector](#)< [data\\_type](#) > \*)

## Additional Inherited Members

### 6.2.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 30 of file analysisdynamic.h.

### 6.2.2 Constructor & Destructor Documentation

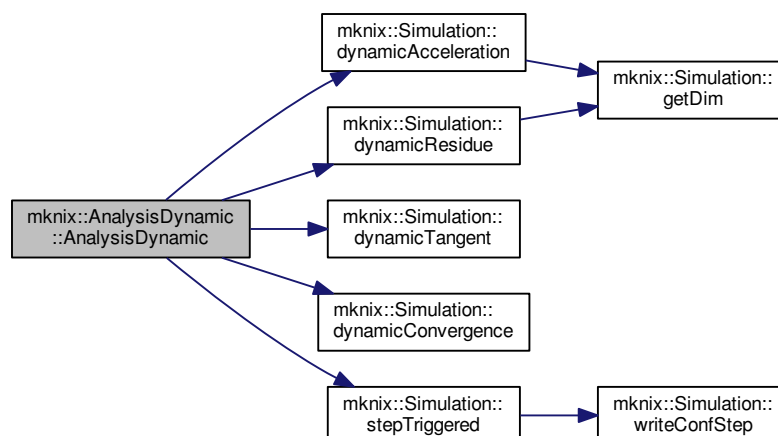
#### 6.2.2.1 mknix::AnalysisDynamic::AnalysisDynamic ( )

Definition at line 25 of file analysisdynamic.cpp.

#### 6.2.2.2 mknix::AnalysisDynamic::AnalysisDynamic ( Simulation \* *simulation\_in*, double *to\_in*, double *tf\_in*, double *At\_in*, char \* *integrator\_in*, double *par1* = -1 ., double *par2* = -1 ., double *par3* = -1 . )

Definition at line 31 of file analysisdynamic.cpp.

Here is the call graph for this function:



#### 6.2.2.3 mknix::AnalysisDynamic::~~AnalysisDynamic ( )

Definition at line 74 of file analysisdynamic.cpp.

### 6.2.3 Member Function Documentation

#### 6.2.3.1 void mknix::AnalysisDynamic::solve ( Imx::Vector< data\_type > \* *q\_in*, Imx::Vector< data\_type > \* *qdot\_in*, Imx::Vector< data\_type > \* *not\_used* = 0 ) [virtual]

Implements [mknix::Analysis](#).

Definition at line 79 of file analysisdynamic.cpp.

Here is the call graph for this function:



6.2.3.2 `std::string mknix::AnalysisDynamic::type ( ) [inline], [virtual]`

Implements [mknix::Analysis](#).

Definition at line 39 of file `analysisdynamic.h`.

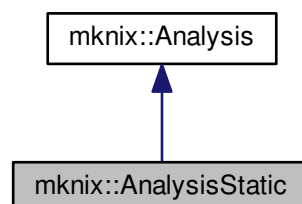
The documentation for this class was generated from the following files:

- [analysisdynamic.h](#)
- [analysisdynamic.cpp](#)

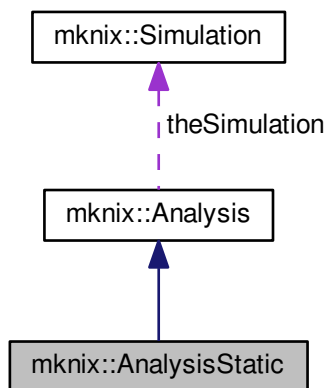
## 6.3 mknix::AnalysisStatic Class Reference

```
#include <analysisstatic.h>
```

Inheritance diagram for `mknix::AnalysisStatic`:



Collaboration diagram for `mknix::AnalysisStatic`:



#### Public Member Functions

- [AnalysisStatic](#) ()
- [AnalysisStatic](#) ([Simulation](#) \*, double)
- [~AnalysisStatic](#) ()
- `std::string` [type](#) ()
- void [solve](#) ([lmx::Vector](#)< [data\\_type](#) > \*, [lmx::Vector](#)< [data\\_type](#) > \*, [lmx::Vector](#)< [data\\_type](#) > \*)

#### Additional Inherited Members

##### 6.3.1 Detailed Description

###### Author

AUTHORS <MAILS>

Definition at line 30 of file `analysisstatic.h`.

##### 6.3.2 Constructor & Destructor Documentation

###### 6.3.2.1 `mknix::AnalysisStatic::AnalysisStatic` ( )

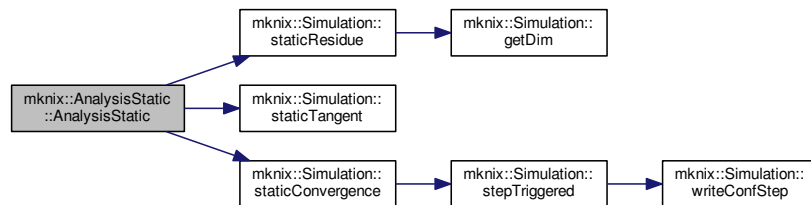
Definition at line 25 of file `analysisstatic.cpp`.

###### 6.3.2.2 `mknix::AnalysisStatic::AnalysisStatic` ( `Simulation` \* *simulation\_in*, double *time\_in* )

Definition at line 31 of file `analysisstatic.cpp`.



Here is the call graph for this function:



### 6.3.2.3 mknix::AnalysisStatic::~~AnalysisStatic ( )

Definition at line 43 of file analysisstatic.cpp.

## 6.3.3 Member Function Documentation

**6.3.3.1** void mknix::AnalysisStatic::solve ( Imx::Vector< data\_type > \* *q\_in*, Imx::Vector< data\_type > \* *qdot\_in* = 0, Imx::Vector< data\_type > \* *not\_used* = 0 ) [virtual]

Implements [mknix::Analysis](#).

Definition at line 48 of file analysisstatic.cpp.

**6.3.3.2** std::string mknix::AnalysisStatic::type ( ) [inline],[virtual]

Implements [mknix::Analysis](#).

Definition at line 39 of file analysisstatic.h.

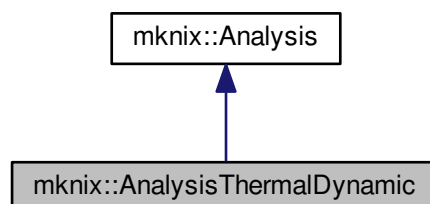
The documentation for this class was generated from the following files:

- [analysisstatic.h](#)
- [analysisstatic.cpp](#)

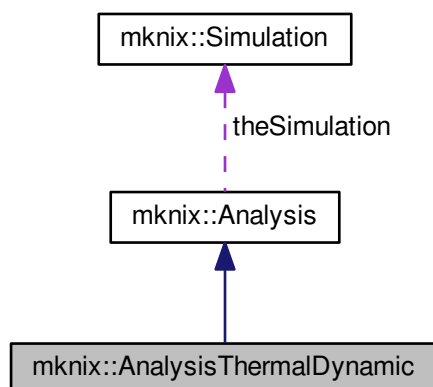
## 6.4 mknix::AnalysisThermalDynamic Class Reference

```
#include <analysisthermaldynamic.h>
```

Inheritance diagram for mknix::AnalysisThermalDynamic:



Collaboration diagram for `mknix::AnalysisThermalDynamic`:



#### Public Member Functions

- [AnalysisThermalDynamic](#) ()
- [AnalysisThermalDynamic](#) ([Simulation](#) \*, double, double, double, char \*)
- [~AnalysisThermalDynamic](#) ()
- `std::string type` ()
- void `init` ([Imx::Vector](#)< `data_type` > \*)
- void `nextStep` ()
- void `solve` ([Imx::Vector](#)< `data_type` > \*, [Imx::Vector](#)< `data_type` > \*, [Imx::Vector](#)< `data_type` > \*)

#### Additional Inherited Members

##### 6.4.1 Detailed Description

###### Author

AUTHORS <MAILS>

Definition at line 30 of file `analysisthermaldynamic.h`.

##### 6.4.2 Constructor & Destructor Documentation

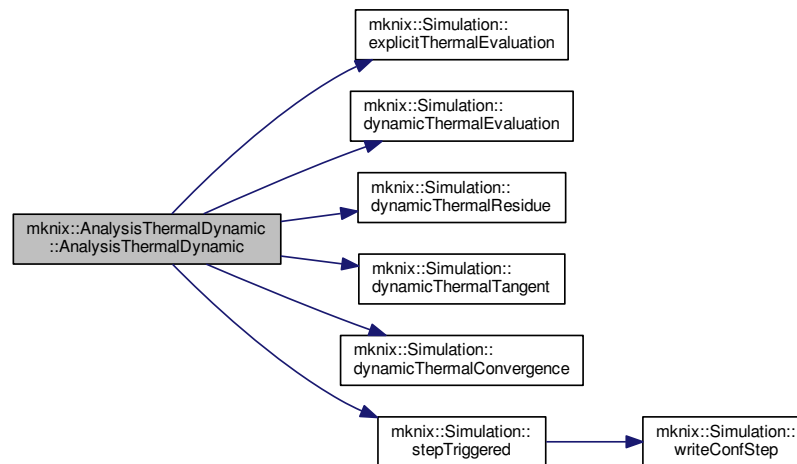
###### 6.4.2.1 `mknix::AnalysisThermalDynamic::AnalysisThermalDynamic` ( )

Definition at line 25 of file `analysisthermaldynamic.cpp`.

###### 6.4.2.2 `mknix::AnalysisThermalDynamic::AnalysisThermalDynamic` ( `Simulation` \* *simulation\_in*, double *to\_in*, double *tf\_in*, double *At\_in*, char \* *integrator\_in* )

Definition at line 31 of file `analysisthermaldynamic.cpp`.

Here is the call graph for this function:



#### 6.4.2.3 mknix::AnalysisThermalDynamic::~~AnalysisThermalDynamic ( )

Definition at line 64 of file analysisthermaldynamic.cpp.

### 6.4.3 Member Function Documentation

#### 6.4.3.1 void mknix::AnalysisThermalDynamic::init ( Imx::Vector< data\_type > \* qt\_in ) [virtual]

Reimplemented from [mknix::Analysis](#).

Definition at line 68 of file analysisthermaldynamic.cpp.

#### 6.4.3.2 void mknix::AnalysisThermalDynamic::nextStep ( ) [virtual]

Reimplemented from [mknix::Analysis](#).

Definition at line 77 of file analysisthermaldynamic.cpp.

#### 6.4.3.3 void mknix::AnalysisThermalDynamic::solve ( Imx::Vector< data\_type > \* qt\_in, Imx::Vector< data\_type > \* qdot\_in = 0, Imx::Vector< data\_type > \* not\_used = 0 ) [virtual]

Implements [mknix::Analysis](#).

Definition at line 83 of file analysisthermaldynamic.cpp.

#### 6.4.3.4 std::string mknix::AnalysisThermalDynamic::type ( ) [inline],[virtual]

Implements [mknix::Analysis](#).

Definition at line 39 of file analysisthermaldynamic.h.

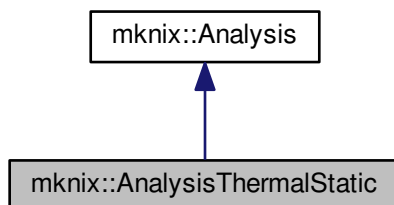
The documentation for this class was generated from the following files:

- [analysisthermaldynamic.h](#)
- [analysisthermaldynamic.cpp](#)

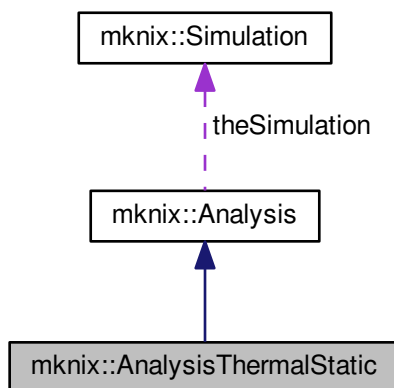
## 6.5 mknix::AnalysisThermalStatic Class Reference

```
#include <analysisthermalstatic.h>
```

Inheritance diagram for mknix::AnalysisThermalStatic:



Collaboration diagram for mknix::AnalysisThermalStatic:



### Public Member Functions

- [AnalysisThermalStatic](#) ()
- [AnalysisThermalStatic](#) ([Simulation](#) \*, double)
- [~AnalysisThermalStatic](#) ()
- `std::string type` ()
- void [solve](#) ([lmx::Vector](#)< [data\\_type](#) > \*, [lmx::Vector](#)< [data\\_type](#) > \*, [lmx::Vector](#)< [data\\_type](#) > \*)

### Additional Inherited Members

#### 6.5.1 Detailed Description

Author

AUTHORS <MAILS>

Definition at line 30 of file analysisthermalstatic.h.

## 6.5.2 Constructor & Destructor Documentation

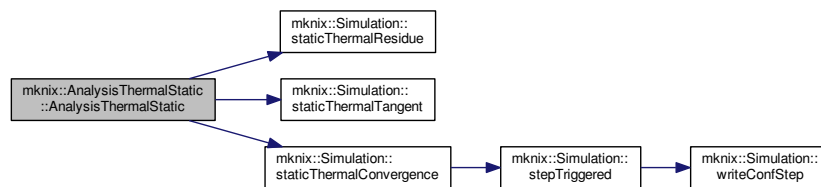
### 6.5.2.1 mknix::AnalysisThermalStatic::AnalysisThermalStatic ( )

Definition at line 25 of file analysisthermalstatic.cpp.

### 6.5.2.2 mknix::AnalysisThermalStatic::AnalysisThermalStatic ( Simulation \* *simulation\_in*, double *time\_in* )

Definition at line 31 of file analysisthermalstatic.cpp.

Here is the call graph for this function:



### 6.5.2.3 mknix::AnalysisThermalStatic::~~AnalysisThermalStatic ( )

Definition at line 43 of file analysisthermalstatic.cpp.

## 6.5.3 Member Function Documentation

### 6.5.3.1 void mknix::AnalysisThermalStatic::solve ( Imx::Vector< data\_type > \* *q\_in*, Imx::Vector< data\_type > \* *qdot\_in* = 0, Imx::Vector< data\_type > \* *not\_used* = 0 ) [virtual]

Implements [mnix::Analysis](#).

Definition at line 48 of file analysisthermalstatic.cpp.

### 6.5.3.2 std::string mknix::AnalysisThermalStatic::type ( ) [inline], [virtual]

Implements [mnix::Analysis](#).

Definition at line 39 of file analysisthermalstatic.h.

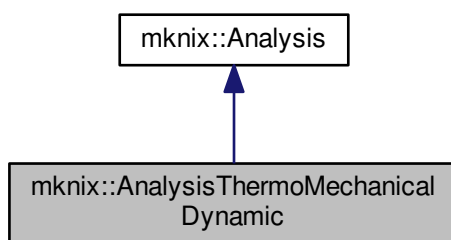
The documentation for this class was generated from the following files:

- [analysisthermalstatic.h](#)
- [analysisthermalstatic.cpp](#)

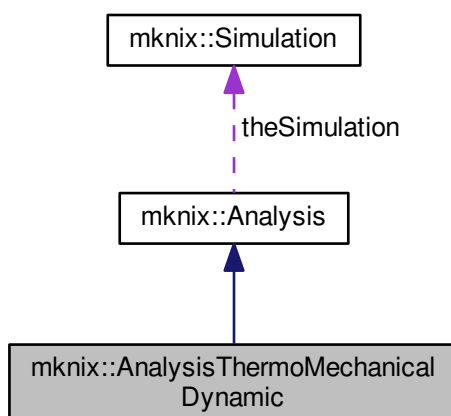
## 6.6 mknix::AnalysisThermoMechanicalDynamic Class Reference

```
#include <analysisthermomechanicaldynamic.h>
```

Inheritance diagram for `mknix::AnalysisThermoMechanicalDynamic`:



Collaboration diagram for `mknix::AnalysisThermoMechanicalDynamic`:



#### Public Member Functions

- [AnalysisThermoMechanicalDynamic](#) ()
- [AnalysisThermoMechanicalDynamic](#) ([Simulation](#) \*, double, double, double, char \*)
- [~AnalysisThermoMechanicalDynamic](#) ()
- `std::string` [type](#) ()
- `void` [solve](#) (`lmx::Vector`< [data\\_type](#) > \*, `lmx::Vector`< [data\\_type](#) > \*, `lmx::Vector`< [data\\_type](#) > \*)

#### Additional Inherited Members

##### 6.6.1 Detailed Description

Author

AUTHORS <MAILS>

Definition at line 32 of file analysisisthermomechanicaldynamic.h.

## 6.6.2 Constructor & Destructor Documentation

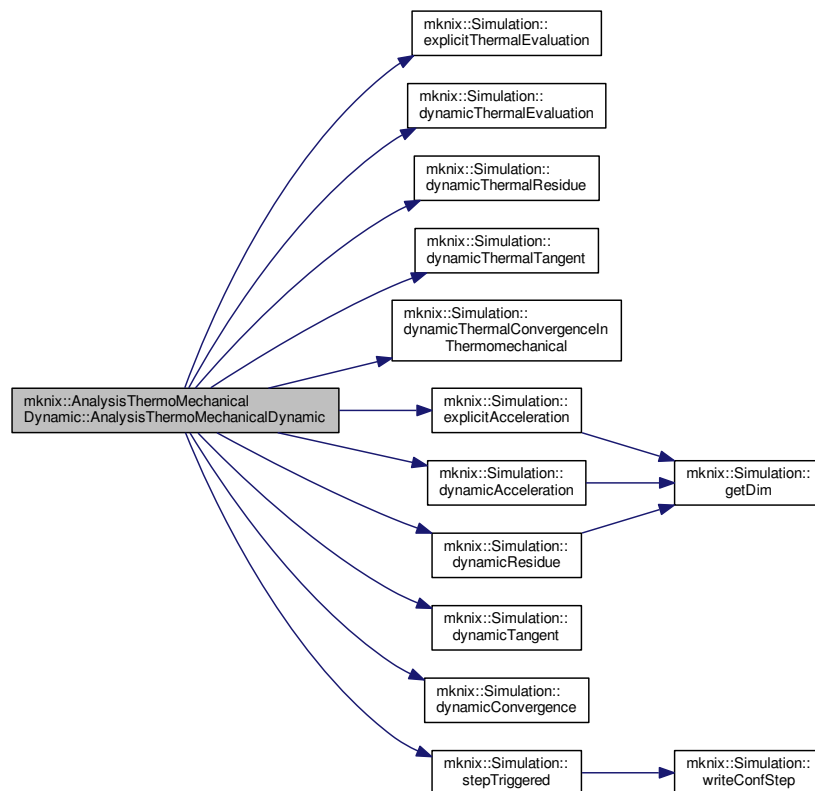
### 6.6.2.1 mknix::AnalysisThermoMechanicalDynamic::AnalysisThermoMechanicalDynamic ( )

Definition at line 25 of file analysisisthermomechanicaldynamic.cpp.

### 6.6.2.2 mknix::AnalysisThermoMechanicalDynamic::AnalysisThermoMechanicalDynamic ( Simulation \* *simulation\_in*, double *to\_in*, double *tf\_in*, double *At\_in*, char \* *integrator\_in* )

Definition at line 32 of file analysisisthermomechanicaldynamic.cpp.

Here is the call graph for this function:



### 6.6.2.3 mknix::AnalysisThermoMechanicalDynamic::~~AnalysisThermoMechanicalDynamic ( )

Definition at line 84 of file analysisisthermomechanicaldynamic.cpp.

## 6.6.3 Member Function Documentation

6.6.3.1 `void mknix::AnalysisThermoMechanicalDynamic::solve ( Imx::Vector< data_type > * qt_in, Imx::Vector< data_type > * q_in, Imx::Vector< data_type > * qdot_in = 0 ) [virtual]`

Implements [mknix::Analysis](#).

Definition at line 90 of file `analysisthermomechanicaldynamic.cpp`.

6.6.3.2 `std::string mknix::AnalysisThermoMechanicalDynamic::type ( ) [inline],[virtual]`

Implements [mknix::Analysis](#).

Definition at line 41 of file `analysisthermomechanicaldynamic.h`.

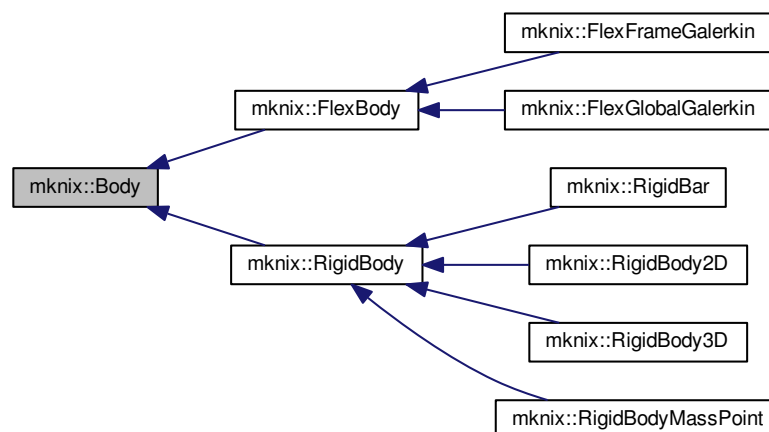
The documentation for this class was generated from the following files:

- [analysisthermomechanicaldynamic.h](#)
- [analysisthermomechanicaldynamic.cpp](#)

## 6.7 mknix::Body Class Reference

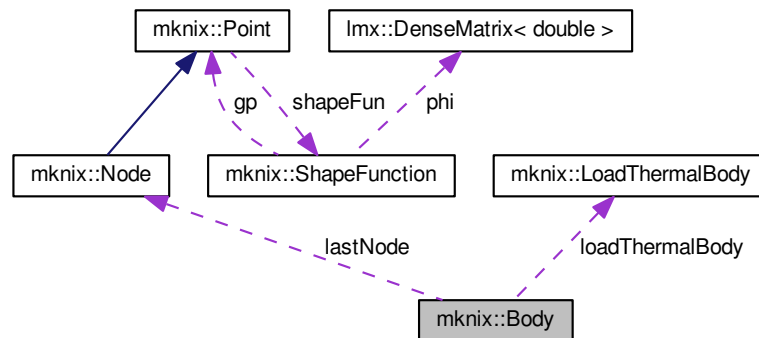
```
#include <body.h>
```

Inheritance diagram for `mknix::Body`:





Collaboration diagram for mknix::Body:



### Public Member Functions

- [Body](#) ()
- [Body](#) (std::string)  
*Constructor with 1 parameter.*
- virtual [~Body](#) ()
- virtual std::string [getType](#) ()=0
- virtual void [initialize](#) ()  
*Cascade initialization funtion. Calls the initialize methods for each of the Cells and tells them to compute their shapefunction values. Both loops are parallelized.*
- virtual void [calcCapacityMatrix](#) ()  
*Computes the local Capacity of the material body by calling each cell's cascade function.*
- virtual void [calcConductivityMatrix](#) ()  
*Computes the local Conductivity of the material body by calling each cell's cascade function.*
- virtual void [calcExternalHeat](#) ()  
*Computes the local volumetric heat vector of the material body by calling each cell's cascade function.*
- virtual void [assembleCapacityMatrix](#) (Imx::Matrix< data\_type > &)  
*Assembles the local conductivity into the global matrix by calling each cell's cascade function.*
- virtual void [assembleConductivityMatrix](#) (Imx::Matrix< data\_type > &)  
*Assembles the local conductivity into the global matrix by calling each cell's cascade function.*
- virtual void [assembleExternalHeat](#) (Imx::Vector< data\_type > &)  
*Assembles the local volumetric heat into the global heat load vector by calling each cell's cascade function.*
- virtual void [calcMassMatrix](#) ()=0
- virtual void [calcExternalForces](#) ()=0
- virtual void [assembleMassMatrix](#) (Imx::Matrix< data\_type > &)=0
- virtual void [assembleExternalForces](#) (Imx::Vector< data\_type > &)=0
- void [setTemperature](#) (double)
- virtual void [setMechanical](#) ()
- virtual void [setOutput](#) (std::string)=0
- virtual void [outputStep](#) (const Imx::Vector< data\_type > &, const Imx::Vector< data\_type > &)=0
- virtual void [outputStep](#) (const Imx::Vector< data\_type > &)=0
- virtual void [outputStep](#) ()  
*Postprocess and store thermal step results for any analysis.*
- virtual void [outputToFile](#) (std::ofstream \*)

*Streams the data stored during the analysis to a file.*

- virtual void `addNode` (`Node *node_in`)
- void `addNodes` (`std::vector< Node * > &nodes_in`)
- virtual int `getNodesSize` ()
- `std::vector< Node * > & getNodes` ()
- virtual `Node * getNode` (int `node_number`)
- virtual `Node * getLastNode` ()
- virtual void `addBoundaryLine` (`Point *node1`, `Point *node2`)
- virtual void `addBoundaryConnectivity` (`std::vector< int > connectivity_in`)
- virtual int `getBoundarySize` ()
- virtual `Point * getBoundaryFirstNode` ()
- virtual `Point * getBoundaryNextNode` (`Point *node_in`)
- void `addBoundaryGroup` (`std::string boundaryName_in`)
- void `addNodeToBoundaryGroup` (int `nodeNumber_in`, `std::string boundaryName_in`)
- void `addCellToBoundaryGroup` (`CellBoundary *cell_in`, `std::string boundaryName_in`)
- void `setLoadThermalInBoundaryGroup` (`LoadThermalBoundary1D *load_in`, `std::string boundaryName_in`)
- int `getCellLastNumber` ()
- virtual void `addCell` (int `num`, `Cell *cell_in`)
- virtual void `writeBodyInfo` (`std::ofstream *`)=0
- virtual void `writeBoundaryNodes` (`std::vector< Point * > &`)=0
- virtual void `writeBoundaryConnectivity` (`std::vector< std::vector< Point * > > &`)=0
- virtual void `translate` (double, double, double)
- virtual void `setLoadThermal` (`LoadThermalBody *theLoad`)

#### Protected Attributes

- `std::string title`
- `Node * lastNode`
- `std::vector< Node * > nodes`
- `std::vector< Node * > bondedBodyNodes`
- `std::map< std::string, BoundaryGroup * > boundaryGroups`
- `std::vector< std::vector< int > > boundaryConnectivity`
- `std::map< Point *, Point * > linearBoundary`
- `std::map< int, Cell * > cells`
- bool `computeEnergy`
- bool `isThermal`
- `std::vector< lmxx::Vector< data_type > * > temperature`
- `LoadThermalBody * loadThermalBody`

#### 6.7.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 46 of file `body.h`.

#### 6.7.2 Constructor & Destructor Documentation

##### 6.7.2.1 `mkniX::Body::Body ( )`

Definition at line 26 of file `body.cpp`.

##### 6.7.2.2 `mkniX::Body::Body ( std::string title_in )`

Constructor with 1 parameter.

## Parameters

<i>title_in</i>	Name of body in the system. Will be the same as the associated material body
-----------------	--

Definition at line 37 of file body.cpp.

### 6.7.2.3 mknix::Body::~~Body ( ) [virtual]

Definition at line 46 of file body.cpp.

## 6.7.3 Member Function Documentation

### 6.7.3.1 void mknix::Body::addBoundaryConnectivity ( std::vector< int > *connectivity\_in* ) [virtual]

Definition at line 311 of file body.cpp.

### 6.7.3.2 void mknix::Body::addBoundaryGroup ( std::string *boundaryName\_in* ) [inline]

Definition at line 162 of file body.h.

### 6.7.3.3 virtual void mknix::Body::addBoundaryLine ( Point \* *node1*, Point \* *node2* ) [inline], [virtual]

Definition at line 128 of file body.h.

### 6.7.3.4 virtual void mknix::Body::addCell ( int *num*, Cell \* *cell\_in* ) [inline], [virtual]

Definition at line 188 of file body.h.

### 6.7.3.5 void mknix::Body::addCellToBoundaryGroup ( CellBoundary \* *cell\_in*, std::string *boundaryName\_in* ) [inline]

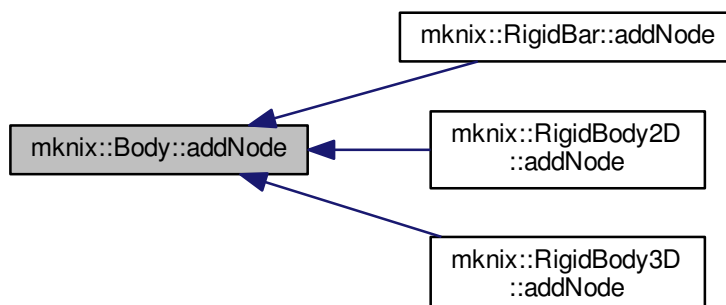
Definition at line 173 of file body.h.

### 6.7.3.6 virtual void mknix::Body::addNode ( Node \* *node\_in* ) [inline], [virtual]

Reimplemented in [mknix::RigidBar](#), [mknix::RigidBody2D](#), and [mknix::RigidBody3D](#).

Definition at line 98 of file body.h.

Here is the caller graph for this function:



6.7.3.7 `void mknix::Body::addNodes ( std::vector< Node * > & nodes_in ) [inline]`

Definition at line 103 of file body.h.

6.7.3.8 `void mknix::Body::addNodeToBoundaryGroup ( int nodeNumber_in, std::string boundaryName_in ) [inline]`

Definition at line 167 of file body.h.

6.7.3.9 `void mknix::Body::assembleCapacityMatrix ( Imx::Matrix< data_type > & globalCapacity ) [virtual]`

Assembles the local conductivity into the global matrix by calling each cell's cascade function.

Parameters

<i>globalCapacity</i>	Reference to the global matrix of the thermal simulation.
-----------------------	---

Returns

void

Definition at line 176 of file body.cpp.

6.7.3.10 `void mknix::Body::assembleConductivityMatrix ( Imx::Matrix< data_type > & globalConductivity ) [virtual]`

Assembles the local conductivity into the global matrix by calling each cell's cascade function.

Parameters

<i>global↔ Conductivity</i>	Reference to the global matrix of the thermal simulation.
---------------------------------	---

Returns

void

Definition at line 191 of file body.cpp.

6.7.3.11 `virtual void mknix::Body::assembleExternalForces ( Imx::Vector< data_type > & ) [pure virtual]`

Implemented in [mknix::FlexFrameGalerkin](#), [mknix::FlexGlobalGalerkin](#), and [mknix::RigidBody](#).

6.7.3.12 `void mknix::Body::assembleExternalHeat ( Imx::Vector< data_type > & globalExternalHeat ) [virtual]`

Assembles the local volumetric heat into the global heat load vector by calling each cell's cascade function.

Returns

void

Definition at line 205 of file body.cpp.

6.7.3.13 `virtual void mknix::Body::assembleMassMatrix ( Imx::Matrix< data_type > & ) [pure virtual]`

Implemented in [mknix::FlexFrameGalerkin](#), [mknix::FlexGlobalGalerkin](#), and [mknix::RigidBody](#).

6.7.3.14 `void mknix::Body::calcCapacityMatrix ( ) [virtual]`

Computes the local Capacity of the material body by calling each cell's cascade function.

Returns

void

Definition at line 128 of file body.cpp.

**6.7.3.15** void mknix::Body::calcConductivityMatrix ( ) [virtual]

Computes the local Conductivity of the material body by calling each cell's cascade function.

**Returns**

void

Definition at line 142 of file body.cpp.

**6.7.3.16** virtual void mknix::Body::calcExternalForces ( ) [pure virtual]

Implemented in [mknix::FlexFrameGalerkin](#), [mknix::FlexGlobalGalerkin](#), [mknix::RigidBar](#), [mknix::RigidBody2D](#), [mknix::RigidBody3D](#), and [mknix::RigidBodyMassPoint](#).

**6.7.3.17** void mknix::Body::calcExternalHeat ( ) [virtual]

Computes the local volumetric heat vector of the material body by calling each cell's cascade function.

**Returns**

void

Definition at line 156 of file body.cpp.

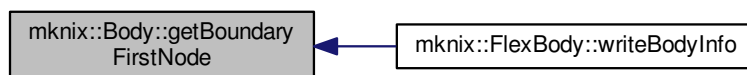
**6.7.3.18** virtual void mknix::Body::calcMassMatrix ( ) [pure virtual]

Implemented in [mknix::FlexFrameGalerkin](#), [mknix::FlexGlobalGalerkin](#), [mknix::RigidBar](#), [mknix::RigidBody2D](#), [mknix::RigidBody3D](#), and [mknix::RigidBodyMassPoint](#).

**6.7.3.19** virtual Point\* mknix::Body::getBoundaryFirstNode ( ) [inline],[virtual]

Definition at line 152 of file body.h.

Here is the caller graph for this function:

**6.7.3.20** virtual Point\* mknix::Body::getBoundaryNextNode ( Point \* node\_in ) [inline],[virtual]

Definition at line 157 of file body.h.

**6.7.3.21** virtual int mknix::Body::getBoundarySize ( ) [inline],[virtual]

Definition at line 147 of file body.h.

Here is the caller graph for this function:



**6.7.3.22** `int mknix::Body::getCellLastNumber ( ) [inline]`

Definition at line 183 of file `body.h`.

**6.7.3.23** `virtual Node* mknix::Body::getLastNode ( ) [inline],[virtual]`

Definition at line 123 of file `body.h`.

Here is the caller graph for this function:



**6.7.3.24** `virtual Node* mknix::Body::getNode ( int node_number ) [inline],[virtual]`

Reimplemented in [mknix::RigidBody](#), and [mknix::FlexBody](#).

Definition at line 118 of file `body.h`.

**6.7.3.25** `std::vector<Node*>& mknix::Body::getNodes ( ) [inline]`

Definition at line 113 of file `body.h`.

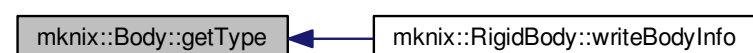
**6.7.3.26** `virtual int mknix::Body::getNodesSize ( ) [inline],[virtual]`

Definition at line 108 of file `body.h`.

**6.7.3.27** `virtual std::string mknix::Body::getType ( ) [pure virtual]`

Implemented in [mknix::RigidBodyMassPoint](#), [mknix::RigidBody](#), [mknix::RigidBody2D](#), [mknix::FlexFrameGalerkin](#), [mknix::FlexGlobalGalerkin](#), and [mknix::RigidBody3D](#).

Here is the caller graph for this function:



### 6.7.3.28 void mknix::Body::initialize ( ) [virtual]

Cascade initialization funtion. Calls the initialize methods for each of the Cells and tells them to compute their shapefunction values. Both loops are parallelized.

#### Returns

void

Reimplemented in [mknix::FlexBody](#).

Definition at line 73 of file body.cpp.

Here is the caller graph for this function:



### 6.7.3.29 virtual void mknix::Body::outputStep ( const Imx::Vector< data\_type > &, const Imx::Vector< data\_type > & ) [pure virtual]

Implemented in [mknix::FlexFrameGalerkin](#), [mknix::FlexGlobalGalerkin](#), and [mknix::RigidBody](#).

### 6.7.3.30 virtual void mknix::Body::outputStep ( const Imx::Vector< data\_type > & ) [pure virtual]

Implemented in [mknix::FlexFrameGalerkin](#), [mknix::FlexGlobalGalerkin](#), and [mknix::RigidBody](#).

### 6.7.3.31 void mknix::Body::outputStep ( ) [virtual]

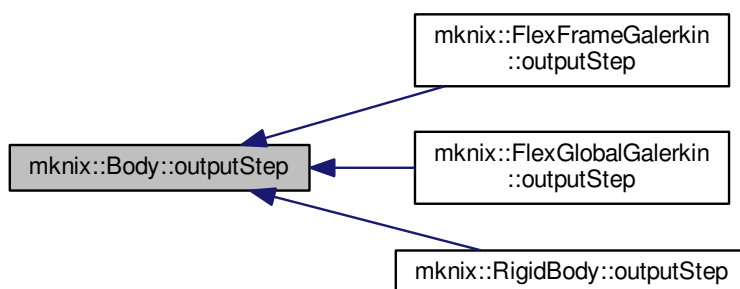
Postprocess and store thermal step results for any analysis.

#### Returns

void

Definition at line 232 of file body.cpp.

Here is the caller graph for this function:



**6.7.3.32** `void mknix::Body::outputToFile ( std::ofstream * outFile ) [virtual]`

Streams the data stored during the analysis to a file.

Parameters

<i>outFile</i>	Output files
----------------	--------------

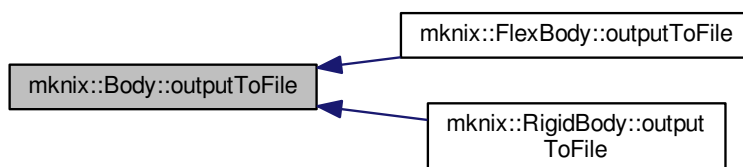
Returns

void

Reimplemented in [mknix::FlexBody](#), and [mknix::RigidBody](#).

Definition at line 267 of file body.cpp.

Here is the caller graph for this function:



**6.7.3.33** `virtual void mknix::Body::setLoadThermal ( LoadThermalBody * theLoad ) [inline],[virtual]`

Definition at line 202 of file body.h.

**6.7.3.34** `void mknix::Body::setLoadThermalInBoundaryGroup ( LoadThermalBoundary1D * load_in, std::string boundaryName_in ) [inline]`

Definition at line 178 of file body.h.

**6.7.3.35** `virtual void mknix::Body::setMechanical ( ) [inline],[virtual]`

Definition at line 82 of file body.h.

**6.7.3.36** `virtual void mknix::Body::setOutput ( std::string ) [pure virtual]`

Implemented in [mknix::FlexBody](#), and [mknix::RigidBody](#).

**6.7.3.37** `void mknix::Body::setTemperature ( double temp_in )`

Definition at line 218 of file body.cpp.

**6.7.3.38** `void mknix::Body::translate ( double x_in, double y_in, double z_in ) [virtual]`

Definition at line 317 of file body.cpp.

**6.7.3.39** `virtual void mknix::Body::writeBodyInfo ( std::ofstream * ) [pure virtual]`

Implemented in [mknix::FlexBody](#), and [mknix::RigidBody](#).



**6.7.3.40** `virtual void mknix::Body::writeBoundaryConnectivity ( std::vector< std::vector< Point * > > & )` [pure virtual]

Implemented in [mknix::FlexBody](#), and [mknix::RigidBody](#).

**6.7.3.41** `virtual void mknix::Body::writeBoundaryNodes ( std::vector< Point * > & )` [pure virtual]

Implemented in [mknix::FlexBody](#), and [mknix::RigidBody](#).

#### 6.7.4 Member Data Documentation

**6.7.4.1** `std::vector<Node *> mknix::Body::bondedBodyNodes` [protected]

Definition at line 212 of file body.h.

**6.7.4.2** `std::vector<std::vector<int> > mknix::Body::boundaryConnectivity` [protected]

Definition at line 214 of file body.h.

**6.7.4.3** `std::map<std::string, BoundaryGroup *> mknix::Body::boundaryGroups` [protected]

Definition at line 213 of file body.h.

**6.7.4.4** `std::map<int, Cell *> mknix::Body::cells` [protected]

Map of integration cells.

Definition at line 217 of file body.h.

**6.7.4.5** `bool mknix::Body::computeEnergy` [protected]

Definition at line 219 of file body.h.

**6.7.4.6** `bool mknix::Body::isThermal` [protected]

Definition at line 220 of file body.h.

**6.7.4.7** `Node* mknix::Body::lastNode` [protected]

Definition at line 210 of file body.h.

**6.7.4.8** `std::map<Point *, Point *> mknix::Body::linearBoundary` [protected]

Map of linear boundary.

Definition at line 215 of file body.h.

**6.7.4.9** `LoadThermalBody* mknix::Body::loadThermalBody` [protected]

Definition at line 222 of file body.h.

**6.7.4.10** `std::vector<Node *> mknix::Body::nodes` [protected]

Definition at line 211 of file body.h.

**6.7.4.11** `std::vector<Imx::Vector<data_type> *> mknix::Body::temperature` [protected]

Definition at line 221 of file body.h.

**6.7.4.12** `std::string mknix::Body::title` [protected]

Definition at line 209 of file body.h.

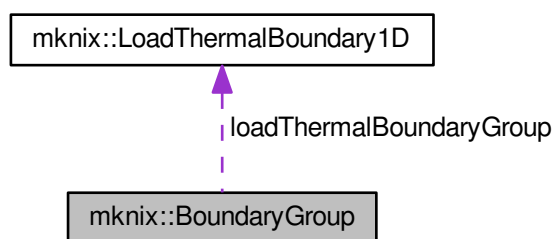
The documentation for this class was generated from the following files:

- [body.h](#)
- [body.cpp](#)

## 6.8 mknix::BoundaryGroup Class Reference

```
#include <boundarygroup.h>
```

Collaboration diagram for mknix::BoundaryGroup:



### Public Member Functions

- [BoundaryGroup](#) ()
- virtual [~BoundaryGroup](#) ()  
*Constructor with 1 parameter.*
- virtual void [initialize](#) ()  
*Cascade initialization funtion. Calls the initialize methods for each of the Cells and tells them to compute their shapefunction values. Both loops are parallelized.*
- virtual void [calcExternalHeat](#) ()  
*Computes the local volumetric heat vector of the material body by calling each cell's cascade function.*
- virtual void [assembleExternalHeat](#) (Imx::Vector< [data\\_type](#) > &)  
*Assembles the local volumetric heat into the global heat load vector by calling each cell's cascade function.*
- virtual void [addNode](#) (Node \*node\_in)
- virtual void [addCell](#) (CellBoundary \*cell\_in)
- virtual void [setLoadThermal](#) (LoadThermalBoundary1D \*theLoad)

### Protected Attributes

- std::vector< [Node](#) \* > [nodes](#)
- std::map< int, [CellBoundary](#) \* > [cells](#)
- [LoadThermalBoundary1D](#) \* [loadThermalBoundaryGroup](#)

#### 6.8.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 40 of file boundarygroup.h.

## 6.8.2 Constructor & Destructor Documentation

### 6.8.2.1 mknix::BoundaryGroup::BoundaryGroup ( )

Definition at line 25 of file boundarygroup.cpp.

### 6.8.2.2 mknix::BoundaryGroup::~BoundaryGroup ( ) [virtual]

Constructor with 1 parameter.

#### Parameters

<i>title_in</i>	Name of boundary in the body.
-----------------	-------------------------------

Definition at line 40 of file boundarygroup.cpp.

## 6.8.3 Member Function Documentation

### 6.8.3.1 virtual void mknix::BoundaryGroup::addCell ( CellBoundary \* *cell\_in* ) [inline], [virtual]

Definition at line 63 of file boundarygroup.h.

### 6.8.3.2 virtual void mknix::BoundaryGroup::addNode ( Node \* *node\_in* ) [inline], [virtual]

Definition at line 58 of file boundarygroup.h.

### 6.8.3.3 void mknix::BoundaryGroup::assembleExternalHeat ( Imx::Vector< data\_type > & *globalExternalHeat* ) [virtual]

Assembles the local volumetric heat into the global heat load vector by calling each cell's cascade function.

#### Returns

void

Definition at line 97 of file boundarygroup.cpp.

### 6.8.3.4 void mknix::BoundaryGroup::calcExternalHeat ( ) [virtual]

Computes the local volumetric heat vector of the material body by calling each cell's cascade function.

#### Returns

void

Definition at line 80 of file boundarygroup.cpp.

### 6.8.3.5 void mknix::BoundaryGroup::initialize ( ) [virtual]

Cascade initialization funtion. Calls the initialize methods for each of the Cells and tells them to compute their shapefunction values. Both loops are parallelized.

#### Returns

void

Definition at line 53 of file boundarygroup.cpp.

### 6.8.3.6 virtual void mknix::BoundaryGroup::setLoadThermal ( LoadThermalBoundary1D \* *theLoad* ) [inline], [virtual]

Definition at line 70 of file boundarygroup.h.

#### 6.8.4 Member Data Documentation

##### 6.8.4.1 `std::map<int, CellBoundary *> mknix::BoundaryGroup::cells` [protected]

Map of integration cells.

Definition at line 78 of file boundarygroup.h.

##### 6.8.4.2 `LoadThermalBoundary1D* mknix::BoundaryGroup::loadThermalBoundaryGroup` [protected]

Definition at line 80 of file boundarygroup.h.

##### 6.8.4.3 `std::vector<Node *> mknix::BoundaryGroup::nodes` [protected]

Definition at line 77 of file boundarygroup.h.

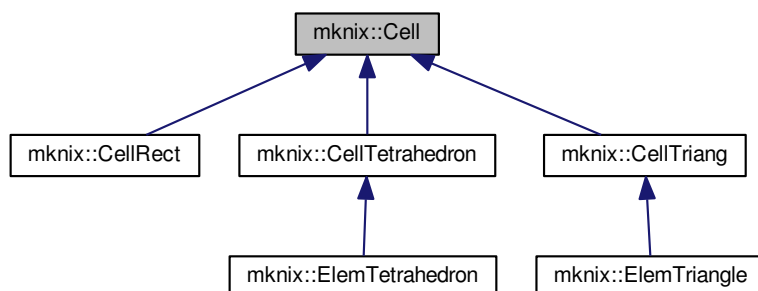
The documentation for this class was generated from the following files:

- [boundarygroup.h](#)
- [boundarygroup.cpp](#)

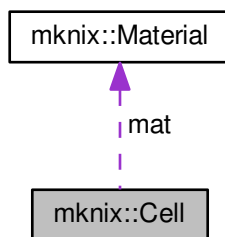
#### 6.9 mknix::Cell Class Reference

```
#include <cell.h>
```

Inheritance diagram for mknix::Cell:



Collaboration diagram for mknix::Cell:



#### Public Member Functions

- [Cell](#) ()
- [Cell](#) ([Material](#) &, std::string, double, int)
- virtual [~Cell](#) ()
- virtual void [initialize](#) (std::vector< [Node](#) \* > &)
- virtual void [computeShapeFunctions](#) ()
- void [computeCapacityGaussPoints](#) ()
- void [assembleCapacityGaussPoints](#) ([Imx::Matrix](#)< [data\\_type](#) > &)
- void [computeConductivityGaussPoints](#) ()
- void [assembleConductivityGaussPoints](#) ([Imx::Matrix](#)< [data\\_type](#) > &)
- void [computeQextGaussPoints](#) ([LoadThermalBody](#) \*)
- void [assembleQextGaussPoints](#) ([Imx::Vector](#)< [data\\_type](#) > &)
- void [computeMGaussPoints](#) ()
- void [assembleMGaussPoints](#) ([Imx::Matrix](#)< [data\\_type](#) > &)
- void [computeFintGaussPoints](#) ()
- void [computeNLFintGaussPoints](#) ()
- void [assembleFintGaussPoints](#) ([Imx::Vector](#)< [data\\_type](#) > &)
- void [computeFextGaussPoints](#) ()
- void [assembleFextGaussPoints](#) ([Imx::Vector](#)< [data\\_type](#) > &)
- void [computeKGaussPoints](#) ()
- void [computeNLKGaussPoints](#) ()
- void [assembleKGaussPoints](#) ([Imx::Matrix](#)< [data\\_type](#) > &)
- void [assembleRGaussPoints](#) ([Imx::Vector](#)< [data\\_type](#) > &, int)
- void [assembleNLRGaussPoints](#) ([Imx::Vector](#)< [data\\_type](#) > &, int)
- double [calcPotentialEGaussPoints](#) (const [Imx::Vector](#)< [data\\_type](#) > &)
- double [calcKineticEGaussPoints](#) (const [Imx::Vector](#)< [data\\_type](#) > &)
- double [calcElasticEGaussPoints](#) ()
- void [outputConnectivityToFile](#) (std::ofstream \*)
- virtual void [gnuplotOut](#) (std::ofstream &, std::ofstream &)=0
- void [gnuplotOutStress](#) (std::ofstream &)

## Protected Attributes

- [Material](#) \* [mat](#)
- std::string [formulation](#)
- double [alpha](#)
- int [nGPoints](#)
- std::vector< [GaussPoint](#) \* > [gPoints](#)
- std::vector< [GaussPoint](#) \* > [gPoints\\_MC](#)
- double [jacobian](#)
- std::vector< [Point](#) \* > [bodyPoints](#)
- double [dc](#)

### 6.9.1 Detailed Description

#### Author

Daniel Iglesias

Definition at line 54 of file cell.h.

### 6.9.2 Constructor & Destructor Documentation

#### 6.9.2.1 `mknix::Cell::Cell ( )`

Definition at line 12 of file cell.cpp.

#### 6.9.2.2 `mknix::Cell::Cell ( Material & material_in, std::string formulation_in, double alpha_in, int nGPoints_in )`

Definition at line 17 of file cell.cpp.

#### 6.9.2.3 `mknix::Cell::~Cell ( )` `[virtual]`

Definition at line 29 of file cell.cpp.

### 6.9.3 Member Function Documentation

#### 6.9.3.1 `void mknix::Cell::assembleCapacityGaussPoints ( Imx::Matrix< data\_type > & globalCapacity )`

Definition at line 87 of file cell.cpp.

#### 6.9.3.2 `void mknix::Cell::assembleConductivityGaussPoints ( Imx::Matrix< data\_type > & globalConductivity )`

Definition at line 102 of file cell.cpp.

#### 6.9.3.3 `void mknix::Cell::assembleFextGaussPoints ( Imx::Vector< data\_type > & globalFext )`

Definition at line 173 of file cell.cpp.

#### 6.9.3.4 `void mknix::Cell::assembleFintGaussPoints ( Imx::Vector< data\_type > & globalFint )`

Definition at line 157 of file cell.cpp.

#### 6.9.3.5 `void mknix::Cell::assembleKGaussPoints ( Imx::Matrix< data\_type > & globalTangent )`

Definition at line 197 of file cell.cpp.

#### 6.9.3.6 `void mknix::Cell::assembleMGaussPoints ( Imx::Matrix< data\_type > & globalMass )`

Definition at line 133 of file cell.cpp.

6.9.3.7 void mknix::Cell::assembleNLRGaussPoints ( Imx::Vector< data\_type > & globalStress, int firstNode )

Definition at line 216 of file cell.cpp.

6.9.3.8 void mknix::Cell::assembleQextGaussPoints ( Imx::Vector< data\_type > & globalQext )

Definition at line 117 of file cell.cpp.

6.9.3.9 void mknix::Cell::assembleRGaussPoints ( Imx::Vector< data\_type > & globalStress, int firstNode )

Definition at line 205 of file cell.cpp.

6.9.3.10 double mknix::Cell::calcElasticEGaussPoints ( )

Definition at line 249 of file cell.cpp.

6.9.3.11 double mknix::Cell::calcKineticEGaussPoints ( const Imx::Vector< data\_type > & qdot )

Definition at line 238 of file cell.cpp.

6.9.3.12 double mknix::Cell::calcPotentialEGaussPoints ( const Imx::Vector< data\_type > & q )

Definition at line 227 of file cell.cpp.

6.9.3.13 void mknix::Cell::computeCapacityGaussPoints ( )

Definition at line 80 of file cell.cpp.

6.9.3.14 void mknix::Cell::computeConductivityGaussPoints ( )

Definition at line 95 of file cell.cpp.

6.9.3.15 void mknix::Cell::computeFextGaussPoints ( )

Definition at line 165 of file cell.cpp.

6.9.3.16 void mknix::Cell::computeFintGaussPoints ( )

Definition at line 141 of file cell.cpp.

6.9.3.17 void mknix::Cell::computeKGaussPoints ( )

Definition at line 181 of file cell.cpp.

6.9.3.18 void mknix::Cell::computeMGaussPoints ( )

Definition at line 125 of file cell.cpp.

6.9.3.19 void mknix::Cell::computeNLFintGaussPoints ( )

Definition at line 149 of file cell.cpp.

6.9.3.20 void mknix::Cell::computeNLKGaussPoints ( )

Definition at line 189 of file cell.cpp.

6.9.3.21 void mknix::Cell::computeQextGaussPoints ( LoadThermalBody \* loadThermalBody\_in )

Definition at line 110 of file cell.cpp.

6.9.3.22 `void mknix::Cell::computeShapeFunctions ( ) [virtual]`

Reimplemented in [mknix::ElemTetrahedron](#), and [mknix::ElemTriangle](#).

Definition at line 68 of file `cell.cpp`.

6.9.3.23 `virtual void mknix::Cell::gnuplotOut ( std::ofstream & , std::ofstream & ) [pure virtual]`

Implemented in [mknix::CellRect](#), [mknix::CellTriang](#), and [mknix::CellTetrahedron](#).

6.9.3.24 `void mknix::Cell::gnuplotOutStress ( std::ofstream & gptension )`

Definition at line 270 of file `cell.cpp`.

6.9.3.25 `void mknix::Cell::initialize ( std::vector< Node * > & nodes_in ) [virtual]`

Reimplemented in [mknix::CellRect](#), [mknix::ElemTetrahedron](#), and [mknix::ElemTriangle](#).

Definition at line 42 of file `cell.cpp`.

6.9.3.26 `void mknix::Cell::outputConnectivityToFile ( std::ofstream * outfile )`

Definition at line 260 of file `cell.cpp`.

#### 6.9.4 Member Data Documentation

6.9.4.1 `double mknix::Cell::alpha [protected]`

Definition at line 118 of file `cell.h`.

6.9.4.2 `std::vector< Point* > mknix::Cell::bodyPoints [protected]`

Definition at line 123 of file `cell.h`.

6.9.4.3 `double mknix::Cell::dc [protected]`

Definition at line 124 of file `cell.h`.

6.9.4.4 `std::string mknix::Cell::formulation [protected]`

Definition at line 117 of file `cell.h`.

6.9.4.5 `std::vector<GaussPoint*> mknix::Cell::gPoints [protected]`

Definition at line 120 of file `cell.h`.

6.9.4.6 `std::vector<GaussPoint*> mknix::Cell::gPoints_MC [protected]`

Definition at line 121 of file `cell.h`.

6.9.4.7 `double mknix::Cell::jacobian [protected]`

Definition at line 122 of file `cell.h`.

6.9.4.8 `Material* mknix::Cell::mat [protected]`

Definition at line 116 of file `cell.h`.

6.9.4.9 `int mknix::Cell::nGPoints [protected]`

number of Gauss Points

Definition at line 119 of file `cell.h`.



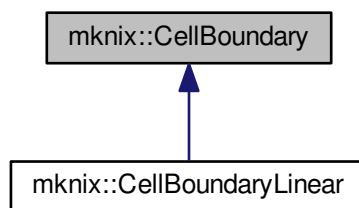
The documentation for this class was generated from the following files:

- [cell.h](#)
- [cell.cpp](#)

## 6.10 mknix::CellBoundary Class Reference

```
#include <cellboundary.h>
```

Inheritance diagram for mknix::CellBoundary:



### Public Member Functions

- [CellBoundary](#) ()
- [CellBoundary](#) (std::string, double, int)
- virtual [~CellBoundary](#) ()
- virtual void [initialize](#) (std::vector< [Node](#) \* > &)
- virtual void [computeShapeFunctions](#) ()
- void [computeQextGaussPoints](#) (LoadThermalBoundary1D \*)
- void [assembleQextGaussPoints](#) (Imx::Vector< [data\\_type](#) > &)

### Protected Attributes

- std::string [formulation](#)
- double [alpha](#)
- int [nGPoints](#)
- std::vector< [GaussPointBoundary](#) \* > [gPoints](#)
- double [jacobian](#)
- std::vector< [Point](#) \* > [bodyPoints](#)
- double [dc](#)

#### 6.10.1 Detailed Description

##### Author

Daniel Iglesias

Definition at line 53 of file cellboundary.h.

## 6.10.2 Constructor & Destructor Documentation

### 6.10.2.1 `mknix::CellBoundary::CellBoundary ( )`

Definition at line 12 of file `cellboundary.cpp`.

### 6.10.2.2 `mknix::CellBoundary::CellBoundary ( std::string formulation_in, double alpha_in, int nGPoints_in )`

Definition at line 17 of file `cellboundary.cpp`.

### 6.10.2.3 `mknix::CellBoundary::~~CellBoundary ( )` `[virtual]`

Definition at line 27 of file `cellboundary.cpp`.

## 6.10.3 Member Function Documentation

### 6.10.3.1 `void mknix::CellBoundary::assembleQextGaussPoints ( Imx::Vector< data_type > & globalQext )`

Definition at line 65 of file `cellboundary.cpp`.

### 6.10.3.2 `void mknix::CellBoundary::computeQextGaussPoints ( LoadThermalBoundary1D * loadThermalBoundary1D_in )`

Definition at line 58 of file `cellboundary.cpp`.

### 6.10.3.3 `void mknix::CellBoundary::computeShapeFunctions ( )` `[virtual]`

Reimplemented in [mknix::CellBoundaryLinear](#).

Definition at line 46 of file `cellboundary.cpp`.

Here is the caller graph for this function:

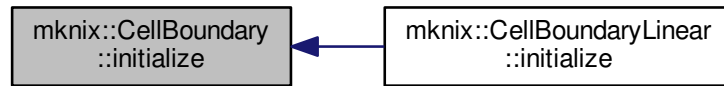


### 6.10.3.4 `void mknix::CellBoundary::initialize ( std::vector< Node * > & nodes_in )` `[virtual]`

Reimplemented in [mknix::CellBoundaryLinear](#).

Definition at line 35 of file `cellboundary.cpp`.

Here is the caller graph for this function:



#### 6.10.4 Member Data Documentation

6.10.4.1 `double mknix::CellBoundary::alpha` [protected]

Definition at line 78 of file `cellboundary.h`.

6.10.4.2 `std::vector< Point* > mknix::CellBoundary::bodyPoints` [protected]

Definition at line 82 of file `cellboundary.h`.

6.10.4.3 `double mknix::CellBoundary::dc` [protected]

Definition at line 83 of file `cellboundary.h`.

6.10.4.4 `std::string mknix::CellBoundary::formulation` [protected]

Definition at line 77 of file `cellboundary.h`.

6.10.4.5 `std::vector<GaussPointBoundary*> mknix::CellBoundary::gPoints` [protected]

Definition at line 80 of file `cellboundary.h`.

6.10.4.6 `double mknix::CellBoundary::jacobian` [protected]

Definition at line 81 of file `cellboundary.h`.

6.10.4.7 `int mknix::CellBoundary::nGPoints` [protected]

number of Gauss Points

Definition at line 79 of file `cellboundary.h`.

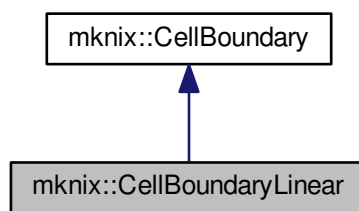
The documentation for this class was generated from the following files:

- [cellboundary.h](#)
- [cellboundary.cpp](#)

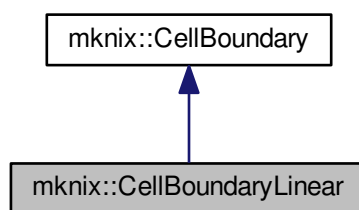
## 6.11 mknix::CellBoundaryLinear Class Reference

```
#include <cellboundarylinear.h>
```

Inheritance diagram for `mknix::CellBoundaryLinear`:



Collaboration diagram for `mknix::CellBoundaryLinear`:



#### Public Member Functions

- [CellBoundaryLinear](#) ()
- [CellBoundaryLinear](#) (std::string, double, int, [Point](#) \*, [Point](#) \*)
- [CellBoundaryLinear](#) (std::string, double, int, [Point](#) \*, [Point](#) \*, double)
- [~CellBoundaryLinear](#) ()
- virtual void [initialize](#) (std::vector< [Node](#) \* > &)
- virtual void [computeShapeFunctions](#) ()

#### Protected Member Functions

- void [createGaussPoints](#) ()

#### Protected Attributes

- `cofe::TensorRank2< 3, double >` [points](#)

#### 6.11.1 Detailed Description

Author

Daniel Iglesias

Definition at line 42 of file cellboundarylinear.h.

### 6.11.2 Constructor & Destructor Documentation

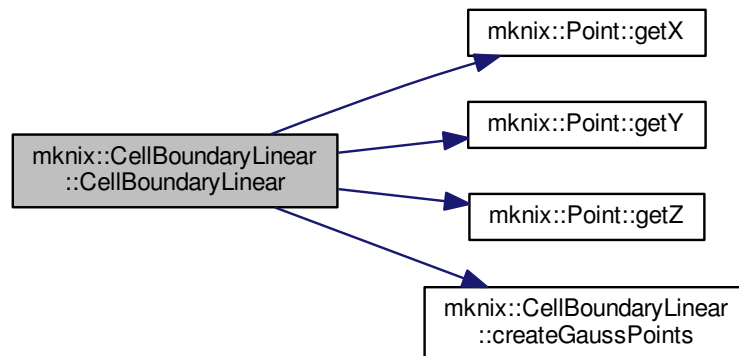
#### 6.11.2.1 mknix::CellBoundaryLinear::CellBoundaryLinear ( )

Definition at line 9 of file cellboundarylinear.cpp.

#### 6.11.2.2 mknix::CellBoundaryLinear::CellBoundaryLinear ( std::string *formulation\_in*, double *alpha\_in*, int *nGPoints\_in*, **Point** \* *n1\_in*, **Point** \* *n2\_in* )

Definition at line 14 of file cellboundarylinear.cpp.

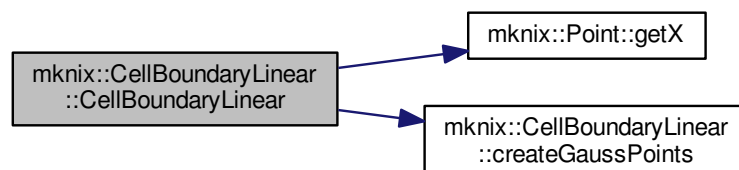
Here is the call graph for this function:



#### 6.11.2.3 mknix::CellBoundaryLinear::CellBoundaryLinear ( std::string *formulation\_in*, double *alpha\_in*, int *nGPoints\_in*, **Point** \* *n1\_in*, **Point** \* *n2\_in*, double *dc\_in* )

Definition at line 42 of file cellboundarylinear.cpp.

Here is the call graph for this function:



#### 6.11.2.4 mknix::CellBoundaryLinear::~~CellBoundaryLinear ( )

Definition at line 70 of file cellboundarylinear.cpp.

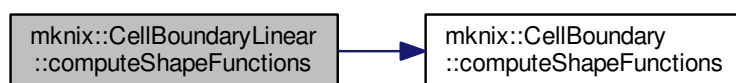
### 6.11.3 Member Function Documentation

#### 6.11.3.1 void mknix::CellBoundaryLinear::computeShapeFunctions ( ) [virtual]

Reimplemented from [mknix::CellBoundary](#).

Definition at line 92 of file cellboundarylinear.cpp.

Here is the call graph for this function:



#### 6.11.3.2 void mknix::CellBoundaryLinear::createGaussPoints ( ) [protected]

Definition at line 105 of file cellboundarylinear.cpp.

Here is the caller graph for this function:

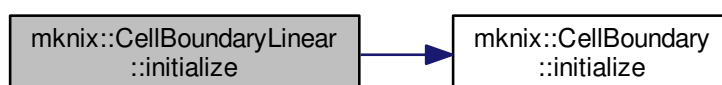


#### 6.11.3.3 void mknix::CellBoundaryLinear::initialize ( std::vector< Node \* > & nodes\_in ) [virtual]

Reimplemented from [mknix::CellBoundary](#).

Definition at line 75 of file cellboundarylinear.cpp.

Here is the call graph for this function:



#### 6.11.4 Member Data Documentation

##### 6.11.4.1 `cofe::TensorRank2<3, double> mknix::CellBoundaryLinear::points` [protected]

position of vertex points

Definition at line 45 of file `cellboundarylinear.h`.

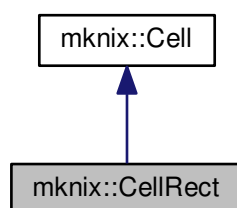
The documentation for this class was generated from the following files:

- [cellboundarylinear.h](#)
- [cellboundarylinear.cpp](#)

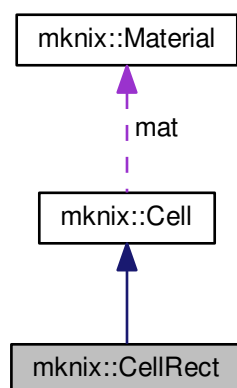
## 6.12 mknix::CellRect Class Reference

```
#include <cellrect.h>
```

Inheritance diagram for `mknix::CellRect`:



Collaboration diagram for `mknix::CellRect`:







6.12.3.2 void mknix::CellRect::initialize ( std::vector< Node \* > & nodes\_in ) [virtual]

Reimplemented from [mknix::Cell](#).

Definition at line 298 of file cellrect.cpp.

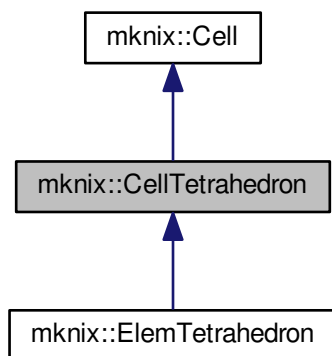
The documentation for this class was generated from the following files:

- [cellrect.h](#)
- [cellrect.cpp](#)

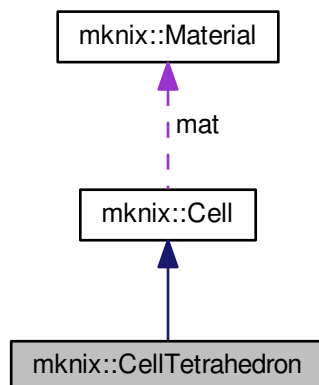
## 6.13 mknix::CellTetrahedron Class Reference

```
#include <celltetrahedron.h>
```

Inheritance diagram for mknix::CellTetrahedron:



Collaboration diagram for mknix::CellTetrahedron:



## Public Member Functions

- [CellTetrahedron](#) ()
- [CellTetrahedron](#) ([Material](#) &, [std::string](#), [double](#), [int](#), [Point](#) \*, [Point](#) \*, [Point](#) \*, [Point](#) \*)
- virtual [~CellTetrahedron](#) ()
- void [gnuplotOut](#) ([std::ofstream](#) &, [std::ofstream](#) &)

## Protected Member Functions

- void [createGaussPoints](#) ()

## Protected Attributes

- [cofe::TensorRank2](#) < 3, [double](#) > [points](#)

### 6.13.1 Detailed Description

#### Author

Daniel Iglesias

Definition at line 43 of file [celltetrahedron.h](#).

### 6.13.2 Constructor & Destructor Documentation

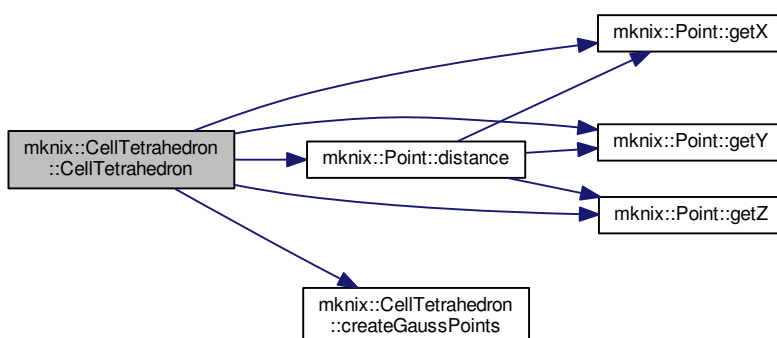
#### 6.13.2.1 [mknix::CellTetrahedron::CellTetrahedron](#) ( )

Definition at line 10 of file [celltetrahedron.cpp](#).

#### 6.13.2.2 [mknix::CellTetrahedron::CellTetrahedron](#) ( [Material](#) & *material\_in*, [std::string](#) *formulation\_in*, [double](#) *alpha\_in*, [int](#) *nGPoints\_in*, [Point](#) \* *n1\_in*, [Point](#) \* *n2\_in*, [Point](#) \* *n3\_in*, [Point](#) \* *n4\_in* )

Definition at line 15 of file [celltetrahedron.cpp](#).

Here is the call graph for this function:



#### 6.13.2.3 [mknix::CellTetrahedron::~~CellTetrahedron](#) ( ) [virtual]

Definition at line 65 of file [celltetrahedron.cpp](#).

## 6.13.3 Member Function Documentation

## 6.13.3.1 void mknix::CellTetrahedron::createGaussPoints ( ) [protected]

Definition at line 70 of file celltetrahedron.cpp.

Here is the caller graph for this function:



## 6.13.3.2 void mknix::CellTetrahedron::gnuplotOut ( std::ofstream &amp; data, std::ofstream &amp; gpdata ) [virtual]

Implements [mknix::Cell](#).

Definition at line 146 of file celltetrahedron.cpp.

## 6.13.4 Member Data Documentation

## 6.13.4.1 cofe::TensorRank2&lt;3, double&gt; mknix::CellTetrahedron::points [protected]

position of vertex points

Definition at line 46 of file celltetrahedron.h.

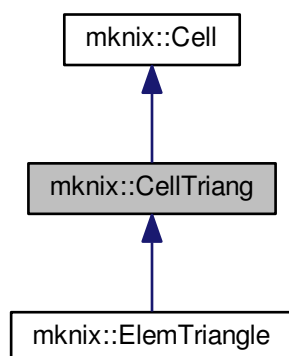
The documentation for this class was generated from the following files:

- [celltetrahedron.h](#)
- [celltetrahedron.cpp](#)

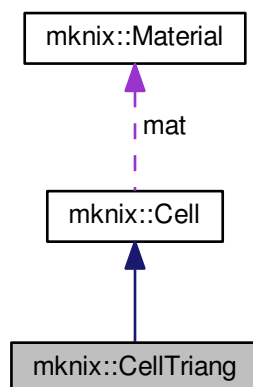
## 6.14 mknix::CellTriang Class Reference

```
#include <celltriang.h>
```

Inheritance diagram for `mknix::CellTriang`:



Collaboration diagram for `mknix::CellTriang`:



#### Public Member Functions

- `CellTriang ()`
- `CellTriang (Material &, std::string, double, int, Point *, Point *, Point *)`
- `CellTriang (Material &, std::string, double, int, Point *, Point *, Point *, double)`
- `virtual ~CellTriang ()`
- `void gnuplotOut (std::ofstream &, std::ofstream &)`

#### Protected Member Functions

- `void createGaussPoints ()`

## Protected Attributes

- `cofe::TensorRank2< 3, double >` [points](#)

## 6.14.1 Detailed Description

## Author

Daniel Iglesias

Definition at line 42 of file `celltriang.h`.

## 6.14.2 Constructor &amp; Destructor Documentation

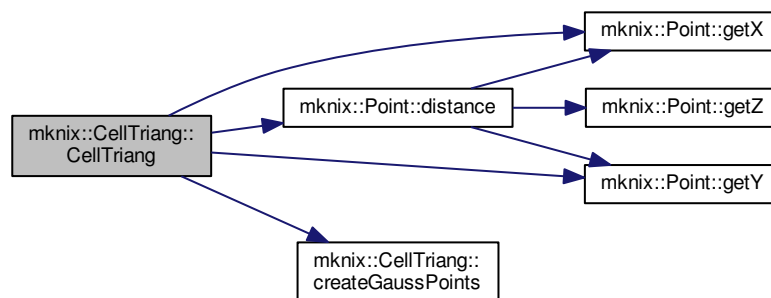
6.14.2.1 `mknix::CellTriang::CellTriang ( )`

Definition at line 9 of file `celltriang.cpp`.

6.14.2.2 `mknix::CellTriang::CellTriang ( Material & material_in, std::string formulation_in, double alpha_in, int nGPoints_in, Point * n1_in, Point * n2_in, Point * n3_in )`

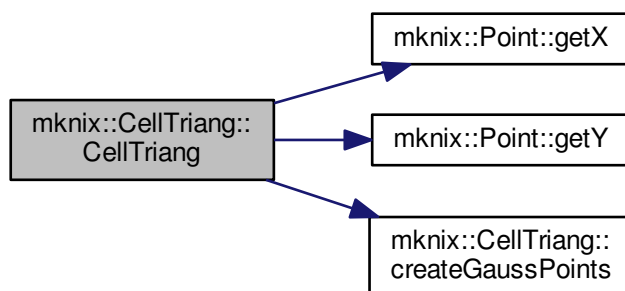
Definition at line 14 of file `celltriang.cpp`.

Here is the call graph for this function:

6.14.2.3 `mknix::CellTriang::CellTriang ( Material & material_in, std::string formulation_in, double alpha_in, int nGPoints_in, Point * n1_in, Point * n2_in, Point * n3_in, double dc_in )`

Definition at line 48 of file `celltriang.cpp`.

Here is the call graph for this function:



#### 6.14.2.4 `mknix::CellTriang::~~CellTriang ( )` [virtual]

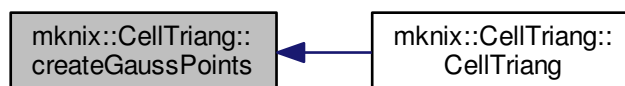
Definition at line 81 of file `celltriang.cpp`.

### 6.14.3 Member Function Documentation

#### 6.14.3.1 `void mknix::CellTriang::createGaussPoints ( )` [protected]

Definition at line 86 of file `celltriang.cpp`.

Here is the caller graph for this function:



#### 6.14.3.2 `void mknix::CellTriang::gnuplotOut ( std::ofstream & data, std::ofstream & gpdata )` [virtual]

Implements [mknix::Cell](#).

Definition at line 288 of file `celltriang.cpp`.

### 6.14.4 Member Data Documentation

#### 6.14.4.1 `cofe::TensorRank2<3, double> mknix::CellTriang::points` [protected]

position of vertex points

Definition at line 45 of file `celltriang.h`.

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

- [celltriang.h](#)
- [celltriang.cpp](#)

## 6.15 mknix::CompBar Class Reference

```
#include <compbar.h>
```

### Public Member Functions

- [CompBar](#) ()
- [CompBar](#) (int, [Node](#) \*, [Node](#) \*)
- [~CompBar](#) ()
- void [addToRender](#) ([vtkRenderer](#) \*)
- void [removeFromRender](#) ([vtkRenderer](#) \*)
- void [updatePoints](#) ()

### 6.15.1 Detailed Description

Definition at line 37 of file compbar.h.

### 6.15.2 Constructor & Destructor Documentation

6.15.2.1 mknix::CompBar::CompBar ( )

6.15.2.2 mknix::CompBar::CompBar ( int , [Node](#) \* , [Node](#) \* )

6.15.2.3 mknix::CompBar::~~CompBar ( )

### 6.15.3 Member Function Documentation

6.15.3.1 void mknix::CompBar::addToRender ( [vtkRenderer](#) \* )

6.15.3.2 void mknix::CompBar::removeFromRender ( [vtkRenderer](#) \* )

6.15.3.3 void mknix::CompBar::updatePoints ( )

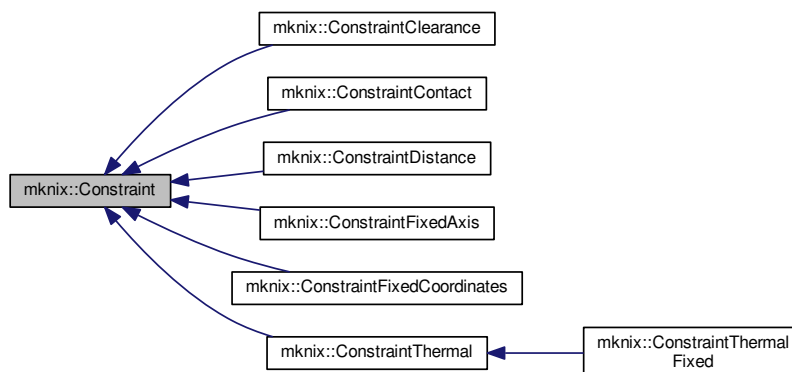
The documentation for this class was generated from the following file:

- [compbar.h](#)

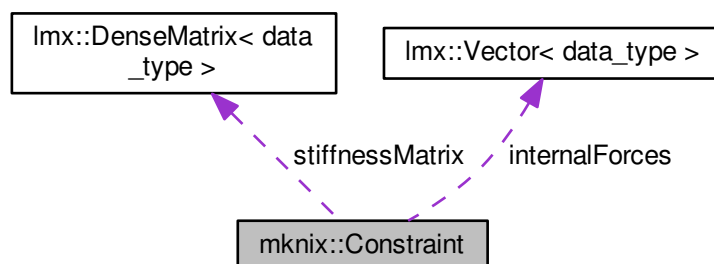
## 6.16 mknix::Constraint Class Reference

```
#include <constraint.h>
```

Inheritance diagram for `mknix::Constraint`:



Collaboration diagram for `mknix::Constraint`:



## Public Member Functions

- `Constraint ()`
- `Constraint (double &, std::string &)`
- `Constraint (double &, std::string &, int)`
- `virtual ~Constraint ()`
- `void setTitle (std::string &title_in)`
- `virtual void writeJointInfo (std::ofstream *)`
- `virtual void calcPhi ()=0`
- `virtual void calcPhiq ()=0`
- `virtual void calcPhiqq ()=0`
- `virtual void calcInternalForces ()`
- `virtual void calcTangentMatrix ()`
- `virtual void assembleInternalForces (Imx::Vector< data_type > &)`
- `virtual void assembleTangentMatrix (Imx::Matrix< data_type > &)`
- `virtual bool checkAugmented ()`
- `virtual void clearAugmented ()`



- virtual [Node](#) \* [getNode](#) (int *nodeNumber*)
- void [outputStep](#) (const [Imx::Vector](#)< [data\\_type](#) > &, const [Imx::Vector](#)< [data\\_type](#) > &)
- void [outputStep](#) (const [Imx::Vector](#)< [data\\_type](#) > &)
- void [outputToFile](#) (std::ofstream \*)

#### Protected Attributes

- int [dim](#)
- double [alpha](#)
- std::string [method](#)
- std::string [title](#)
- std::vector< [Node](#) \* > [nodes](#)
- [Imx::Vector](#)< [data\\_type](#) > [internalForces](#)
- std::vector< [Imx::Vector](#)< [data\\_type](#) > > [internalForcesOutput](#)
- [Imx::DenseMatrix](#)< [data\\_type](#) > [stiffnessMatrix](#)
- std::vector< double > [lambda](#)
- std::vector< double > [phi](#)
- std::vector< [Imx::Vector](#)< [data\\_type](#) > > [phi\\_q](#)
- std::vector< [Imx::DenseMatrix](#)< [data\\_type](#) > > [phi\\_qq](#)

#### 6.16.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 32 of file `constraint.h`.

#### 6.16.2 Constructor & Destructor Documentation

##### 6.16.2.1 mknix::Constraint::Constraint ( )

Definition at line 27 of file `constraint.cpp`.

##### 6.16.2.2 mknix::Constraint::Constraint ( double & *alpha\_in*, std::string & *method\_in* )

Definition at line 31 of file `constraint.cpp`.

##### 6.16.2.3 mknix::Constraint::Constraint ( double & *alpha\_in*, std::string & *method\_in*, int *dim\_in* )

Definition at line 39 of file `constraint.cpp`.

##### 6.16.2.4 mknix::Constraint::~~Constraint ( ) [virtual]

Definition at line 48 of file `constraint.cpp`.

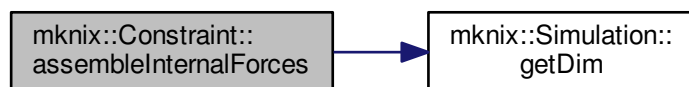
#### 6.16.3 Member Function Documentation

##### 6.16.3.1 void mknix::Constraint::assembleInternalForces ( [Imx::Vector](#)< [data\\_type](#) > & *globalInternalForces* ) [virtual]

Reimplemented in [mknix::ConstraintThermal](#).

Definition at line 102 of file `constraint.cpp`.

Here is the call graph for this function:

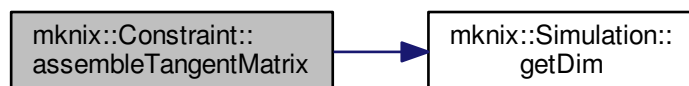


**6.16.3.2** `void mknix::Constraint::assembleTangentMatrix ( Imx::Matrix< data_type > & globalTangent ) [virtual]`

Reimplemented in [mknix::ConstraintThermal](#).

Definition at line 129 of file `constraint.cpp`.

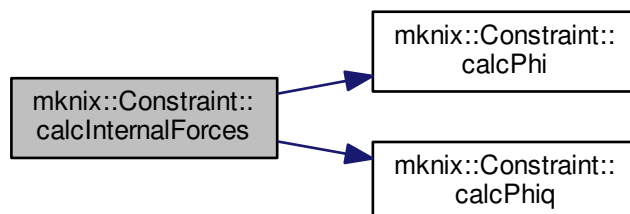
Here is the call graph for this function:



**6.16.3.3** `void mknix::Constraint::calcInternalForces ( ) [virtual]`

Definition at line 67 of file `constraint.cpp`.

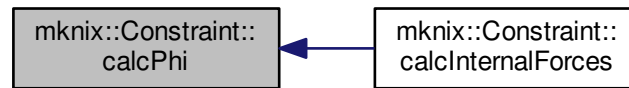
Here is the call graph for this function:



**6.16.3.4** `virtual void mknix::Constraint::calcPhi ( ) [pure virtual]`

Implemented in [mknix::ConstraintDistance](#), [mknix::ConstraintClearance](#), [mknix::ConstraintContact](#), [mknix::ConstraintFixedAxis](#), [mknix::ConstraintFixedCoordinates](#), and [mknix::ConstraintThermalFixed](#).

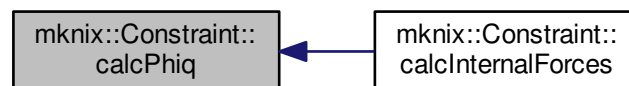
Here is the caller graph for this function:



#### 6.16.3.5 `virtual void mknix::Constraint::calcPhiq ( ) [pure virtual]`

Implemented in [mknix::ConstraintDistance](#), [mknix::ConstraintClearance](#), [mknix::ConstraintContact](#), [mknix::ConstraintFixedAxis](#), [mknix::ConstraintFixedCoordinates](#), and [mknix::ConstraintThermalFixed](#).

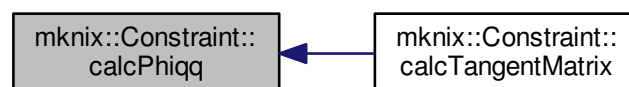
Here is the caller graph for this function:



#### 6.16.3.6 `virtual void mknix::Constraint::calcPhiqq ( ) [pure virtual]`

Implemented in [mknix::ConstraintDistance](#), [mknix::ConstraintClearance](#), [mknix::ConstraintContact](#), [mknix::ConstraintFixedAxis](#), [mknix::ConstraintFixedCoordinates](#), and [mknix::ConstraintThermalFixed](#).

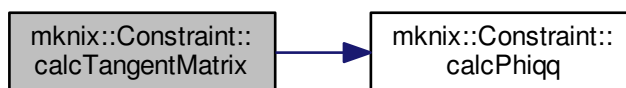
Here is the caller graph for this function:



#### 6.16.3.7 `void mknix::Constraint::calcTangentMatrix ( ) [virtual]`

Definition at line 80 of file `constraint.cpp`.

Here is the call graph for this function:



**6.16.3.8** `bool mknix::Constraint::checkAugmented ( ) [virtual]`

Definition at line 170 of file constraint.cpp.

**6.16.3.9** `void mknix::Constraint::clearAugmented ( ) [virtual]`

Definition at line 195 of file constraint.cpp.

**6.16.3.10** `virtual Node* mknix::Constraint::getNode ( int nodeNumber ) [inline], [virtual]`

Definition at line 65 of file constraint.h.

**6.16.3.11** `void mknix::Constraint::outputStep ( const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot )`

Definition at line 218 of file constraint.cpp.

**6.16.3.12** `void mknix::Constraint::outputStep ( const Imx::Vector< data_type > & q )`

Definition at line 226 of file constraint.cpp.

**6.16.3.13** `void mknix::Constraint::outputToFile ( std::ofstream * outFile )`

Definition at line 234 of file constraint.cpp.

**6.16.3.14** `void mknix::Constraint::setTitle ( std::string & title_in ) [inline]`

Definition at line 42 of file constraint.h.

**6.16.3.15** `void mknix::Constraint::writeJointInfo ( std::ofstream * outfile ) [virtual]`

Definition at line 53 of file constraint.cpp.

## 6.16.4 Member Data Documentation

**6.16.4.1** `double mknix::Constraint::alpha [protected]`

Definition at line 78 of file constraint.h.

**6.16.4.2** `int mknix::Constraint::dim [protected]`

Definition at line 77 of file constraint.h.

**6.16.4.3** `Imx::Vector< data_type > mknix::Constraint::internalForces [protected]`

Definition at line 82 of file constraint.h.

6.16.4.4 `std::vector< Imx::Vector<data_type> > mknix::Constraint::internalForcesOutput` [protected]

Definition at line 83 of file constraint.h.

6.16.4.5 `std::vector< double > mknix::Constraint::lambda` [protected]

Definition at line 85 of file constraint.h.

6.16.4.6 `std::string mknix::Constraint::method` [protected]

Definition at line 79 of file constraint.h.

6.16.4.7 `std::vector<Node*> mknix::Constraint::nodes` [protected]

Definition at line 81 of file constraint.h.

6.16.4.8 `std::vector< double > mknix::Constraint::phi` [protected]

Definition at line 86 of file constraint.h.

6.16.4.9 `std::vector< Imx::Vector<data_type> > mknix::Constraint::phi_q` [protected]

Definition at line 87 of file constraint.h.

6.16.4.10 `std::vector< Imx::DenseMatrix<data_type> > mknix::Constraint::phi_qq` [protected]

Definition at line 88 of file constraint.h.

6.16.4.11 `Imx::DenseMatrix<data_type> mknix::Constraint::stiffnessMatrix` [protected]

Definition at line 84 of file constraint.h.

6.16.4.12 `std::string mknix::Constraint::title` [protected]

Definition at line 80 of file constraint.h.

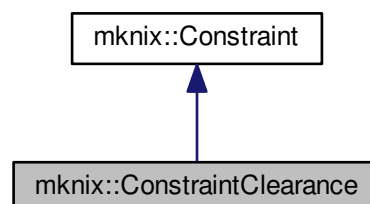
The documentation for this class was generated from the following files:

- [constraint.h](#)
- [constraint.cpp](#)

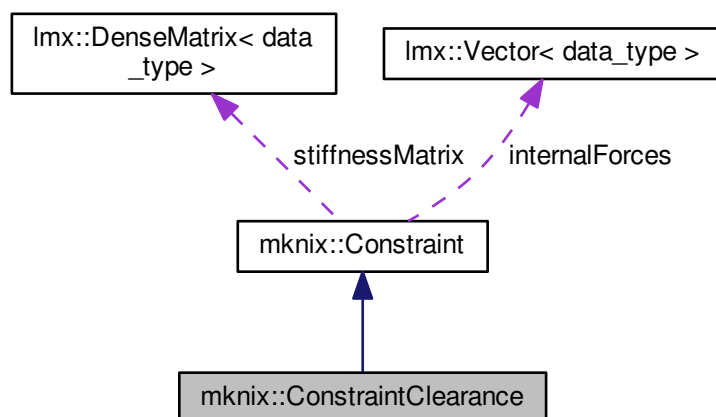
## 6.17 mknix::ConstraintClearance Class Reference

```
#include <constraintclearance.h>
```

Inheritance diagram for mknix::ConstraintClearance:



Collaboration diagram for `mknix::ConstraintClearance`:



#### Public Member Functions

- [ConstraintClearance \(\)](#)
- [~ConstraintClearance \(\)](#)
- [ConstraintClearance \(Node \\*, Node \\*, double &, double &, std::string &\)](#)
- void [calcPhi \(\)](#)
- void [calcPhiq \(\)](#)
- void [calcPhiqq \(\)](#)
- [Imx::Vector< data\\_type > & getInternalForces \(\)](#)
- [Imx::DenseMatrix< data\\_type > & getStiffnessMatrix \(\)](#)

#### Protected Attributes

- double [rh](#)
- double [rt](#)

#### 6.17.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 30 of file `constraintclearance.h`.

#### 6.17.2 Constructor & Destructor Documentation

##### 6.17.2.1 `mknix::ConstraintClearance::ConstraintClearance ( )`

Definition at line 28 of file `constraintclearance.cpp`.

##### 6.17.2.2 `mknix::ConstraintClearance::~~ConstraintClearance ( )`

Definition at line 51 of file `constraintclearance.cpp`.

6.17.2.3 `mknix::ConstraintClearance::ConstraintClearance ( Node * a_in, Node * b_in, double & rh_in, double & alpha_in, std::string & method_in )`

Definition at line 33 of file `constraintclearance.cpp`.

### 6.17.3 Member Function Documentation

6.17.3.1 `void mknix::ConstraintClearance::calcPhi ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 55 of file `constraintclearance.cpp`.

6.17.3.2 `void mknix::ConstraintClearance::calcPhiq ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 64 of file `constraintclearance.cpp`.

6.17.3.3 `void mknix::ConstraintClearance::calcPhiqq ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 92 of file `constraintclearance.cpp`.

6.17.3.4 `Imx::Vector<data_type>& mknix::ConstraintClearance::getInternalForces ( ) [inline]`

Definition at line 45 of file `constraintclearance.h`.

6.17.3.5 `Imx::DenseMatrix<data_type>& mknix::ConstraintClearance::getStiffnessMatrix ( ) [inline]`

Definition at line 49 of file `constraintclearance.h`.

### 6.17.4 Member Data Documentation

6.17.4.1 `double mknix::ConstraintClearance::rh [protected]`

Definition at line 54 of file `constraintclearance.h`.

6.17.4.2 `double mknix::ConstraintClearance::rt [protected]`

Definition at line 55 of file `constraintclearance.h`.

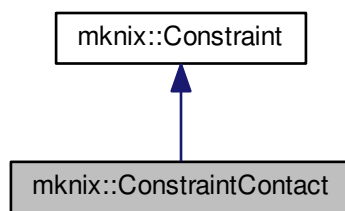
The documentation for this class was generated from the following files:

- [constraintclearance.h](#)
- [constraintclearance.cpp](#)

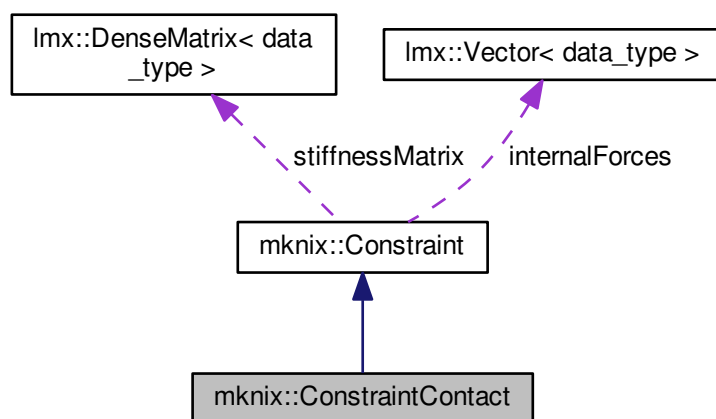
## 6.18 mknix::ConstraintContact Class Reference

```
#include <constraintcontact.h>
```

Inheritance diagram for `mknix::ConstraintContact`:



Collaboration diagram for `mknix::ConstraintContact`:



#### Public Member Functions

- [ConstraintContact](#) ()
- [~ConstraintContact](#) ()
- [ConstraintContact](#) ([Node](#) \*, [Node](#) \*, [Node](#) \*, double &, std::string &)
- void [calcPhi](#) ()
- void [calcPhiq](#) ()
- void [calcPhiqq](#) ()
- [Imx::Vector< data\\_type >](#) & [getInternalForces](#) ()
- [Imx::DenseMatrix< data\\_type >](#) & [getStiffnessMatrix](#) ()
- double [getGap](#) ()

#### Protected Attributes

- std::vector< double > [normal](#)



- double [rh](#)
- double [rt](#)

### 6.18.1 Detailed Description

Author

AUTHORS <MAILS>

Definition at line 30 of file constraintcontact.h.

### 6.18.2 Constructor & Destructor Documentation

#### 6.18.2.1 mknix::ConstraintContact::ConstraintContact ( )

Definition at line 27 of file constraintcontact.cpp.

#### 6.18.2.2 mknix::ConstraintContact::~~ConstraintContact ( )

Definition at line 51 of file constraintcontact.cpp.

#### 6.18.2.3 mknix::ConstraintContact::ConstraintContact ( Node \* *q1\_in*, Node \* *q2\_in*, Node \* *p\_in*, double & *alpha\_in*, std::string & *method\_in* )

Definition at line 32 of file constraintcontact.cpp.

### 6.18.3 Member Function Documentation

#### 6.18.3.1 void mknix::ConstraintContact::calcPhi ( ) [virtual]

Implements [mknix::Constraint](#).

Definition at line 55 of file constraintcontact.cpp.

#### 6.18.3.2 void mknix::ConstraintContact::calcPhiq ( ) [virtual]

Implements [mknix::Constraint](#).

Definition at line 73 of file constraintcontact.cpp.

#### 6.18.3.3 void mknix::ConstraintContact::calcPhiqq ( ) [virtual]

Implements [mknix::Constraint](#).

Definition at line 110 of file constraintcontact.cpp.

#### 6.18.3.4 double mknix::ConstraintContact::getGap ( ) [inline]

Definition at line 53 of file constraintcontact.h.

#### 6.18.3.5 Imx::Vector<data\_type>& mknix::ConstraintContact::getInternalForces ( ) [inline]

Definition at line 45 of file constraintcontact.h.

#### 6.18.3.6 Imx::DenseMatrix<data\_type>& mknix::ConstraintContact::getStiffnessMatrix ( ) [inline]

Definition at line 49 of file constraintcontact.h.

### 6.18.4 Member Data Documentation

#### 6.18.4.1 `std::vector<double> mknix::ConstraintContact::normal` `[protected]`

Definition at line 58 of file `constraintcontact.h`.

#### 6.18.4.2 `double mknix::ConstraintContact::rh` `[protected]`

Definition at line 59 of file `constraintcontact.h`.

#### 6.18.4.3 `double mknix::ConstraintContact::rt` `[protected]`

Definition at line 60 of file `constraintcontact.h`.

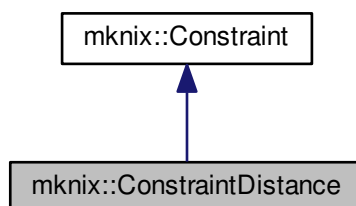
The documentation for this class was generated from the following files:

- [constraintcontact.h](#)
- [constraintcontact.cpp](#)

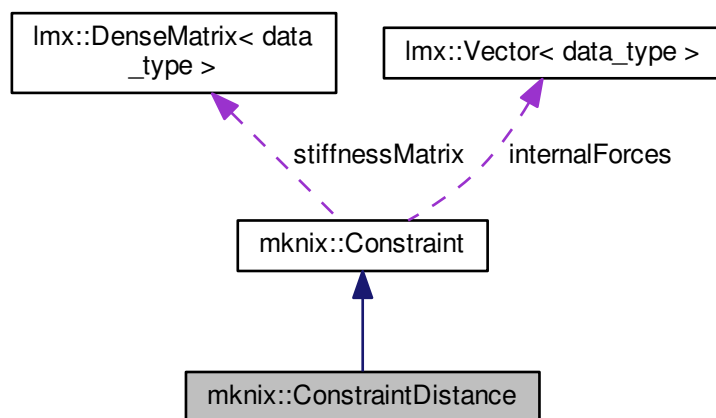
## 6.19 `mknix::ConstraintDistance` Class Reference

```
#include <constraintdistance.h>
```

Inheritance diagram for `mknix::ConstraintDistance`:



Collaboration diagram for mknix::ConstraintDistance:



#### Public Member Functions

- [ConstraintDistance](#) ()
- [~ConstraintDistance](#) ()
- [ConstraintDistance](#) ([Node](#) \*, [Node](#) \*, double &, std::string &)
- void [calcRo](#) ()
- void [calcPhi](#) ()
- void [calcPhiq](#) ()
- void [calcPhiqq](#) ()
- void [setLenght](#) (double new\_ro)
- [Imx::Vector< data\\_type >](#) & [getInternalForces](#) ()
- [Imx::DenseMatrix< data\\_type >](#) & [getStiffnessMatrix](#) ()

#### Protected Attributes

- double [ro](#)
- double [rt](#)

#### 6.19.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 30 of file constraintdistance.h.

#### 6.19.2 Constructor & Destructor Documentation

##### 6.19.2.1 mknix::ConstraintDistance::ConstraintDistance ( )

Definition at line 27 of file constraintdistance.cpp.

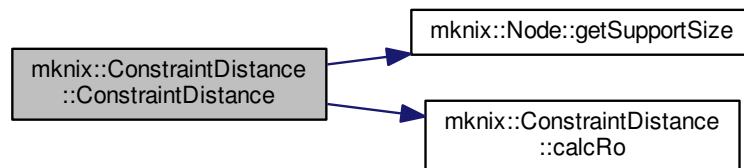
### 6.19.2.2 `mknix::ConstraintDistance::~~ConstraintDistance ( )`

Definition at line 60 of file `constraintdistance.cpp`.

### 6.19.2.3 `mknix::ConstraintDistance::ConstraintDistance ( Node * a_in, Node * b_in, double & alpha_in, std::string & method_in )`

Definition at line 32 of file `constraintdistance.cpp`.

Here is the call graph for this function:



## 6.19.3 Member Function Documentation

### 6.19.3.1 `void mknix::ConstraintDistance::calcPhi ( )` [virtual]

Implements [mknix::Constraint](#).

Definition at line 76 of file `constraintdistance.cpp`.

### 6.19.3.2 `void mknix::ConstraintDistance::calcPhiq ( )` [virtual]

Implements [mknix::Constraint](#).

Definition at line 106 of file `constraintdistance.cpp`.

### 6.19.3.3 `void mknix::ConstraintDistance::calcPhiqq ( )` [virtual]

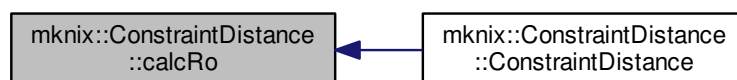
Implements [mknix::Constraint](#).

Definition at line 118 of file `constraintdistance.cpp`.

### 6.19.3.4 `void mknix::ConstraintDistance::calcRo ( )`

Definition at line 64 of file `constraintdistance.cpp`.

Here is the caller graph for this function:



6.19.3.5 `Imx::Vector<data_type>& mknix::ConstraintDistance::getInternalForces ( ) [inline]`

Definition at line 52 of file `constraintdistance.h`.

6.19.3.6 `Imx::DenseMatrix<data_type>& mknix::ConstraintDistance::getStiffnessMatrix ( ) [inline]`

Definition at line 56 of file `constraintdistance.h`.

6.19.3.7 `void mknix::ConstraintDistance::setLenght ( double new_ro ) [inline]`

Definition at line 47 of file `constraintdistance.h`.

#### 6.19.4 Member Data Documentation

6.19.4.1 `double mknix::ConstraintDistance::ro [protected]`

Definition at line 61 of file `constraintdistance.h`.

6.19.4.2 `double mknix::ConstraintDistance::rt [protected]`

Definition at line 62 of file `constraintdistance.h`.

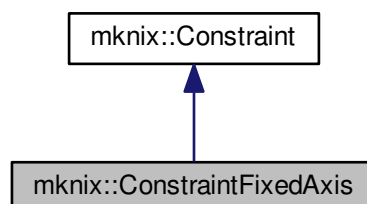
The documentation for this class was generated from the following files:

- [constraintdistance.h](#)
- [constraintdistance.cpp](#)

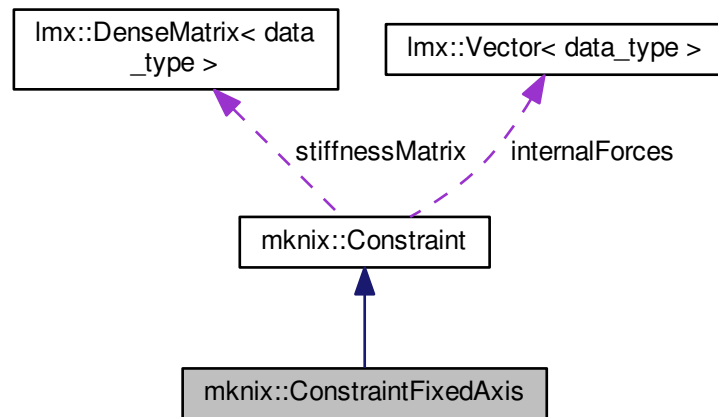
## 6.20 mknix::ConstraintFixedAxis Class Reference

```
#include <constraintfixedaxis.h>
```

Inheritance diagram for `mknix::ConstraintFixedAxis`:



Collaboration diagram for `mknix::ConstraintFixedAxis`:



#### Public Member Functions

- [ConstraintFixedAxis](#) ()
- [ConstraintFixedAxis](#) ([Node](#) \*, [Node](#) \*, [std::string](#) &, [double](#) &, [std::string](#) &)
- [~ConstraintFixedAxis](#) ()
- void [calcPhi](#) ()
- void [calcPhiq](#) ()
- void [calcPhiqq](#) ()
- [Imx::Vector< data\\_type >](#) & [getInternalForces](#) ()
- [Imx::DenseMatrix< data\\_type >](#) & [getStiffnessMatrix](#) ()

#### Protected Attributes

- [std::string](#) [axisName](#)
- [double](#) [ro](#)
- [double](#) [rt](#)

#### 6.20.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 30 of file `constraintfixedaxis.h`.

#### 6.20.2 Constructor & Destructor Documentation

##### 6.20.2.1 `mknix::ConstraintFixedAxis::ConstraintFixedAxis ( )`

Definition at line 27 of file `constraintfixedaxis.cpp`.

6.20.2.2 `mknix::ConstraintFixedAxis::ConstraintFixedAxis ( Node * a_in, Node * b_in, std::string & axisName_in, double & alpha_in, std::string & method_in )`

Definition at line 33 of file `constraintfixedaxis.cpp`.

Here is the call graph for this function:



6.20.2.3 `mknix::ConstraintFixedAxis::~~ConstraintFixedAxis ( )`

Definition at line 64 of file `constraintfixedaxis.cpp`.

### 6.20.3 Member Function Documentation

6.20.3.1 `void mknix::ConstraintFixedAxis::calcPhi ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 68 of file `constraintfixedaxis.cpp`.

6.20.3.2 `void mknix::ConstraintFixedAxis::calcPhiq ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 80 of file `constraintfixedaxis.cpp`.

6.20.3.3 `void mknix::ConstraintFixedAxis::calcPhiqq ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 133 of file `constraintfixedaxis.cpp`.

6.20.3.4 `Imx::Vector<data_type>& mknix::ConstraintFixedAxis::getInternalForces ( ) [inline]`

Definition at line 45 of file `constraintfixedaxis.h`.

6.20.3.5 `Imx::DenseMatrix<data_type>& mknix::ConstraintFixedAxis::getStiffnessMatrix ( ) [inline]`

Definition at line 49 of file `constraintfixedaxis.h`.

### 6.20.4 Member Data Documentation

6.20.4.1 `std::string mknix::ConstraintFixedAxis::axisName [protected]`

Definition at line 54 of file `constraintfixedaxis.h`.

6.20.4.2 `double mknix::ConstraintFixedAxis::ro [protected]`

Definition at line 55 of file `constraintfixedaxis.h`.

#### 6.20.4.3 `double mknix::ConstraintFixedAxis::rt` [protected]

Definition at line 56 of file `constraintfixedaxis.h`.

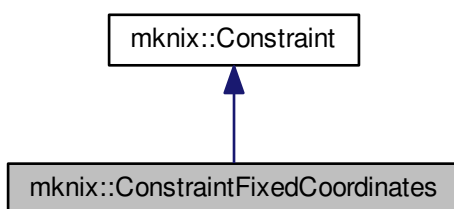
The documentation for this class was generated from the following files:

- [constraintfixedaxis.h](#)
- [constraintfixedaxis.cpp](#)

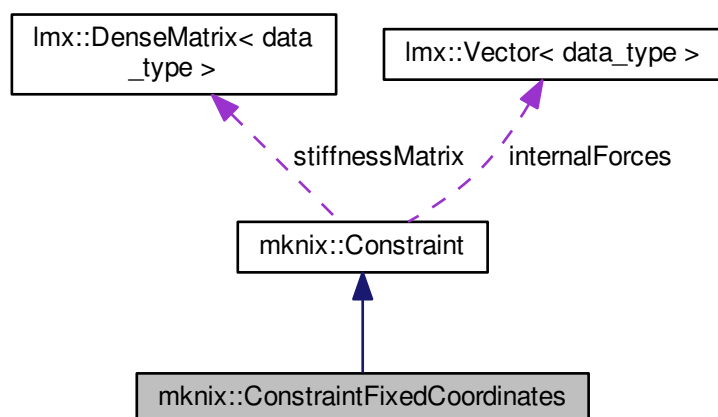
### 6.21 `mknix::ConstraintFixedCoordinates` Class Reference

```
#include <constraintfixedcoordinates.h>
```

Inheritance diagram for `mknix::ConstraintFixedCoordinates`:



Collaboration diagram for `mknix::ConstraintFixedCoordinates`:



#### Public Member Functions

- [ConstraintFixedCoordinates](#) ()
- [ConstraintFixedCoordinates](#) ([Node](#) \*, [Node](#) \*, double &, std::string &)



- [~ConstraintFixedCoordinates](#) ()
- void [calcPhi](#) ()
- void [calcPhiq](#) ()
- void [calcPhiqq](#) ()
- [Imx::Vector](#)< [data\\_type](#) > & [getInternalForces](#) ()
- [Imx::DenseMatrix](#)< [data\\_type](#) > & [getStiffnessMatrix](#) ()

#### Protected Attributes

- double [rxo](#)
- double [ryo](#)
- double [rzo](#)
- double [rxt](#)
- double [ryt](#)
- double [rzt](#)

#### 6.21.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 30 of file `constraintfixedcoordinates.h`.

#### 6.21.2 Constructor & Destructor Documentation

##### 6.21.2.1 mknix::ConstraintFixedCoordinates::ConstraintFixedCoordinates ( )

Definition at line 27 of file `constraintfixedcoordinates.cpp`.

##### 6.21.2.2 mknix::ConstraintFixedCoordinates::ConstraintFixedCoordinates ( **Node** \* *a\_in*, **Node** \* *b\_in*, double & *alpha\_in*, **std::string** & *method\_in* )

Definition at line 33 of file `constraintfixedcoordinates.cpp`.

Here is the call graph for this function:



##### 6.21.2.3 mknix::ConstraintFixedCoordinates::~~ConstraintFixedCoordinates ( )

Definition at line 74 of file `constraintfixedcoordinates.cpp`.

#### 6.21.3 Member Function Documentation

##### 6.21.3.1 void mknix::ConstraintFixedCoordinates::calcPhi ( ) [virtual]

Implements [mknix::Constraint](#).

Definition at line 78 of file `constraintfixedcoordinates.cpp`.

6.21.3.2 `void mknix::ConstraintFixedCoordinates::calcPhiq ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 94 of file `constraintfixedcoordinates.cpp`.

6.21.3.3 `void mknix::ConstraintFixedCoordinates::calcPhiqq ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 121 of file `constraintfixedcoordinates.cpp`.

6.21.3.4 `Imx::Vector<data_type>& mknix::ConstraintFixedCoordinates::getInternalForces ( ) [inline]`

Definition at line 45 of file `constraintfixedcoordinates.h`.

6.21.3.5 `Imx::DenseMatrix<data_type>& mknix::ConstraintFixedCoordinates::getStiffnessMatrix ( ) [inline]`

Definition at line 49 of file `constraintfixedcoordinates.h`.

#### 6.21.4 Member Data Documentation

6.21.4.1 `double mknix::ConstraintFixedCoordinates::rxo [protected]`

Definition at line 54 of file `constraintfixedcoordinates.h`.

6.21.4.2 `double mknix::ConstraintFixedCoordinates::rxt [protected]`

Definition at line 55 of file `constraintfixedcoordinates.h`.

6.21.4.3 `double mknix::ConstraintFixedCoordinates::ryo [protected]`

Definition at line 54 of file `constraintfixedcoordinates.h`.

6.21.4.4 `double mknix::ConstraintFixedCoordinates::ryt [protected]`

Definition at line 55 of file `constraintfixedcoordinates.h`.

6.21.4.5 `double mknix::ConstraintFixedCoordinates::rzo [protected]`

Definition at line 54 of file `constraintfixedcoordinates.h`.

6.21.4.6 `double mknix::ConstraintFixedCoordinates::rzt [protected]`

Definition at line 55 of file `constraintfixedcoordinates.h`.

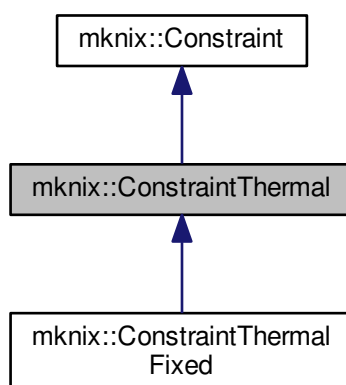
The documentation for this class was generated from the following files:

- [constraintfixedcoordinates.h](#)
- [constraintfixedcoordinates.cpp](#)

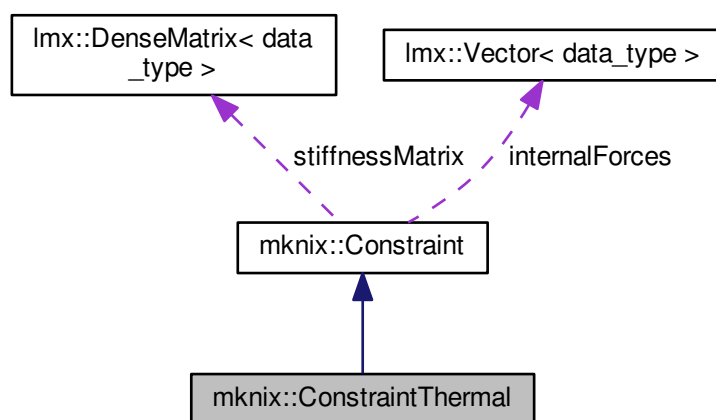
## 6.22 mknix::ConstraintThermal Class Reference

```
#include <constraintthermal.h>
```

Inheritance diagram for mknix::ConstraintThermal:



Collaboration diagram for mknix::ConstraintThermal:



#### Public Member Functions

- [ConstraintThermal](#) ()
- [ConstraintThermal](#) (double &, std::string &)
- [~ConstraintThermal](#) ()
- virtual void [assembleInternalForces](#) (Imx::Vector< data\_type > &)
- virtual void [assembleTangentMatrix](#) (Imx::Matrix< data\_type > &)

#### Additional Inherited Members

### 6.22.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 34 of file `constraintthermal.h`.

### 6.22.2 Constructor & Destructor Documentation

#### 6.22.2.1 `mknix::ConstraintThermal::ConstraintThermal ( )`

Definition at line 27 of file `constraintthermal.cpp`.

#### 6.22.2.2 `mknix::ConstraintThermal::ConstraintThermal ( double & alpha_in, std::string & method_in )`

Definition at line 33 of file `constraintthermal.cpp`.

#### 6.22.2.3 `mknix::ConstraintThermal::~~ConstraintThermal ( )`

Definition at line 39 of file `constraintthermal.cpp`.

### 6.22.3 Member Function Documentation

#### 6.22.3.1 `void mknix::ConstraintThermal::assembleInternalForces ( Imx::Vector< data_type > & globalInternalForces )` [virtual]

Reimplemented from [mknix::Constraint](#).

Definition at line 44 of file `constraintthermal.cpp`.

#### 6.22.3.2 `void mknix::ConstraintThermal::assembleTangentMatrix ( Imx::Matrix< data_type > & globalTangent )` [virtual]

Reimplemented from [mknix::Constraint](#).

Definition at line 70 of file `constraintthermal.cpp`.

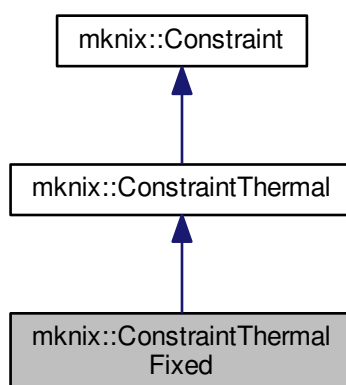
The documentation for this class was generated from the following files:

- [constraintthermal.h](#)
- [constraintthermal.cpp](#)

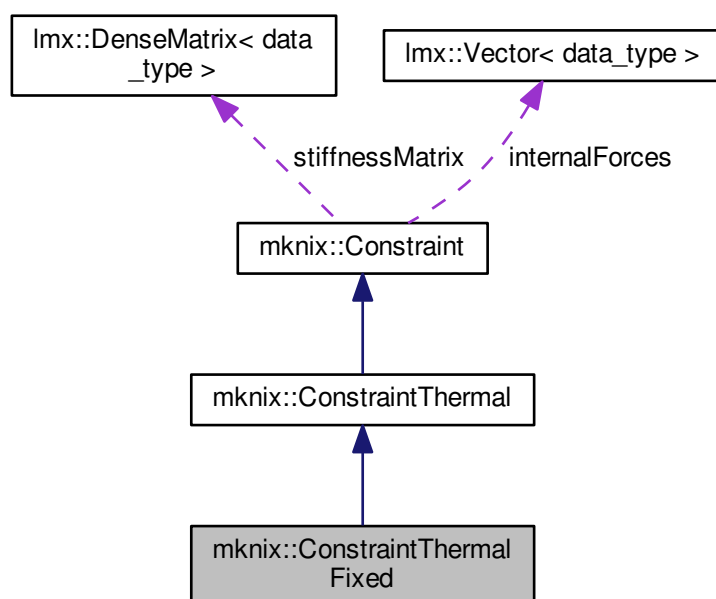
## 6.23 `mknix::ConstraintThermalFixed` Class Reference

```
#include <constraintthermalfixed.h>
```

Inheritance diagram for mknix::ConstraintThermalFixed:



Collaboration diagram for mknix::ConstraintThermalFixed:



#### Public Member Functions

- [ConstraintThermalFixed](#) ()
- [ConstraintThermalFixed](#) (Node \*, Node \*, double &, std::string &)
- [~ConstraintThermalFixed](#) ()

- void `calcPhi` ()
- void `calcPhiq` ()
- void `calcPhiqq` ()
- `Imx::Vector< data_type > & getInternalForces` ()
- `Imx::DenseMatrix< data_type > & getStiffnessMatrix` ()

#### Protected Attributes

- double `To`
- double `Tt`

#### 6.23.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 30 of file `constraintthermalfixed.h`.

#### 6.23.2 Constructor & Destructor Documentation

##### 6.23.2.1 `mknix::ConstraintThermalFixed::ConstraintThermalFixed ( )`

Definition at line 27 of file `constraintthermalfixed.cpp`.

##### 6.23.2.2 `mknix::ConstraintThermalFixed::ConstraintThermalFixed ( Node * a_in, Node * b_in, double & alpha_in, std::string & method_in )`

Definition at line 33 of file `constraintthermalfixed.cpp`.

Here is the call graph for this function:



##### 6.23.2.3 `mknix::ConstraintThermalFixed::~~ConstraintThermalFixed ( )`

Definition at line 63 of file `constraintthermalfixed.cpp`.

#### 6.23.3 Member Function Documentation

##### 6.23.3.1 `void mknix::ConstraintThermalFixed::calcPhi ( )` [virtual]

Implements `mknix::Constraint`.

Definition at line 67 of file `constraintthermalfixed.cpp`.

6.23.3.2 `void mknix::ConstraintThermalFixed::calcPhiq ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 77 of file `constraintthermalfixed.cpp`.

6.23.3.3 `void mknix::ConstraintThermalFixed::calcPhiqq ( ) [virtual]`

Implements [mknix::Constraint](#).

Definition at line 98 of file `constraintthermalfixed.cpp`.

6.23.3.4 `Imx::Vector<data_type>& mknix::ConstraintThermalFixed::getInternalForces ( ) [inline]`

Definition at line 45 of file `constraintthermalfixed.h`.

6.23.3.5 `Imx::DenseMatrix<data_type>& mknix::ConstraintThermalFixed::getStiffnessMatrix ( ) [inline]`

Definition at line 49 of file `constraintthermalfixed.h`.

## 6.23.4 Member Data Documentation

6.23.4.1 `double mknix::ConstraintThermalFixed::To [protected]`

Definition at line 54 of file `constraintthermalfixed.h`.

6.23.4.2 `double mknix::ConstraintThermalFixed::Tt [protected]`

Definition at line 55 of file `constraintthermalfixed.h`.

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

- [constraintthermalfixed.h](#)
- [constraintthermalfixed.cpp](#)

## 6.24 mknix::Contact Class Reference

```
#include <generalcontact.h>
```

### Public Member Functions

- [Contact](#) ()
- [Contact](#) ([Simulation](#) \*, double)
- [~Contact](#) ()
- void [createPoints](#) ()
- void [updatePoints](#) ()
- void [createPolys](#) ()
- void [updateLines](#) ()
- void [createDelaunay](#) ()
- void [updateDelaunay](#) ()
- void [createDrawingObjects](#) ()
- void [createDrawingContactObjects](#) ()
- void [drawObjects](#) ()

### 6.24.1 Detailed Description

Definition at line 50 of file `generalcontact.h`.

### 6.24.2 Constructor & Destructor Documentation

6.24.2.1 `mknix::Contact::Contact ( )`

6.24.2.2 `mknix::Contact::Contact ( Simulation *, double )`

6.24.2.3 `mknix::Contact::~~Contact ( )` `[inline]`

Definition at line 54 of file `generalcontact.h`.

### 6.24.3 Member Function Documentation

6.24.3.1 `void mknix::Contact::createDelaunay ( )`

6.24.3.2 `void mknix::Contact::createDrawingContactObjects ( )`

6.24.3.3 `void mknix::Contact::createDrawingObjects ( )`

6.24.3.4 `void mknix::Contact::createPoints ( )`

6.24.3.5 `void mknix::Contact::createPolys ( )`

6.24.3.6 `void mknix::Contact::drawObjects ( )`

6.24.3.7 `void mknix::Contact::updateDelaunay ( )`

6.24.3.8 `void mknix::Contact::updateLines ( )`

6.24.3.9 `void mknix::Contact::updatePoints ( )`

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

- [generalcontact.h](#)

## 6.25 Imx::DenseMatrix< T > Class Template Reference

```
#include <cellrect.h>
```

### 6.25.1 Detailed Description

```
template<typename T>class Imx::DenseMatrix< T >
```

Definition at line 39 of file `cellrect.h`.

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

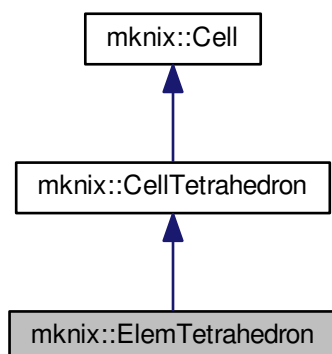
- [cellrect.h](#)

## 6.26 mknix::ElemTetrahedron Class Reference

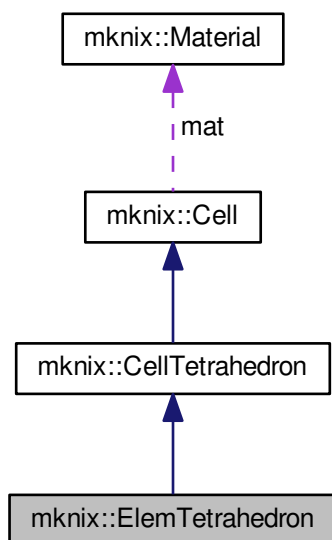
```
#include <elemtetrahedron.h>
```



Inheritance diagram for mknix::ElemTetrahedron:



Collaboration diagram for mknix::ElemTetrahedron:



#### Public Member Functions

- [ElemTetrahedron](#) ()
- [ElemTetrahedron](#) ([Material](#) &, double, int, [Node](#) \*, [Node](#) \*, [Node](#) \*, [Node](#) \*)
- [~ElemTetrahedron](#) ()
- void [initialize](#) (std::vector< [Node](#) \* > &)
- void [computeShapeFunctions](#) ()
- void [createGaussPoints\\_MC](#) ()

## Additional Inherited Members

### 6.26.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 30 of file elemtetrahedron.h.

### 6.26.2 Constructor & Destructor Documentation

#### 6.26.2.1 mknix::ElemTetrahedron::ElemTetrahedron ( )

Definition at line 27 of file elemtetrahedron.cpp.

#### 6.26.2.2 mknix::ElemTetrahedron::ElemTetrahedron ( **Material & material\_in**, **double alpha\_in**, **int nGPoints\_in**, **Node \* n1\_in**, **Node \* n2\_in**, **Node \* n3\_in**, **Node \* n4\_in** )

Definition at line 33 of file elemtetrahedron.cpp.

#### 6.26.2.3 mknix::ElemTetrahedron::~ElemTetrahedron ( )

Definition at line 52 of file elemtetrahedron.cpp.

### 6.26.3 Member Function Documentation

#### 6.26.3.1 void mknix::ElemTetrahedron::computeShapeFunctions ( ) [virtual]

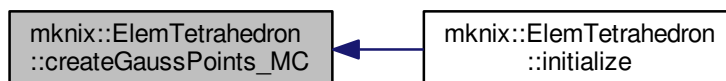
Reimplemented from [mknix::Cell](#).

Definition at line 78 of file elemtetrahedron.cpp.

#### 6.26.3.2 void mknix::ElemTetrahedron::createGaussPoints\_MC ( )

Definition at line 88 of file elemtetrahedron.cpp.

Here is the caller graph for this function:



#### 6.26.3.3 void mknix::ElemTetrahedron::initialize ( **std::vector< Node \* > & nodes\_in** ) [virtual]

Reimplemented from [mknix::Cell](#).

Definition at line 57 of file elemtetrahedron.cpp.

Here is the call graph for this function:



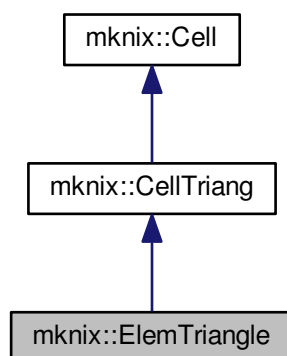
The documentation for this class was generated from the following files:

- [elemtetrahedron.h](#)
- [elemtetrahedron.cpp](#)

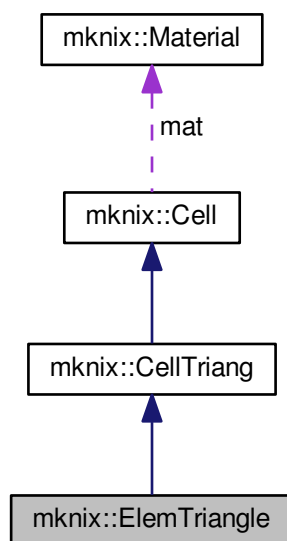
## 6.27 mknix::ElemTriangle Class Reference

```
#include <elemtriangle.h>
```

Inheritance diagram for `mknix::ElemTriangle`:



Collaboration diagram for `mknix::ElemTriangle`:



#### Public Member Functions

- [ElemTriangle](#) ()
- [ElemTriangle](#) ([Material](#) &, double, int, [Node](#) \*, [Node](#) \*, [Node](#) \*)
- [~ElemTriangle](#) ()
- void [initialize](#) (std::vector< [Node](#) \* > &)
- void [computeShapeFunctions](#) ()

#### Protected Member Functions

- void [createGaussPoints\\_MC](#) ()

#### Additional Inherited Members

##### 6.27.1 Detailed Description

###### Author

AUTHORS <MAILS>

Definition at line 30 of file `elemtriangle.h`.

##### 6.27.2 Constructor & Destructor Documentation

###### 6.27.2.1 `mknix::ElemTriangle::ElemTriangle ( )`

Definition at line 27 of file `elemtriangle.cpp`.

6.27.2.2 `mknix::ElemTriangle::ElemTriangle ( Material & material_in, double alpha_in, int nGPoints_in, Node * n1_in, Node * n2_in, Node * n3_in )`

Definition at line 33 of file `elemtriangle.cpp`.

6.27.2.3 `mknix::ElemTriangle::~ElemTriangle ( )`

Definition at line 50 of file `elemtriangle.cpp`.

### 6.27.3 Member Function Documentation

6.27.3.1 `void mknix::ElemTriangle::computeShapeFunctions ( )` [virtual]

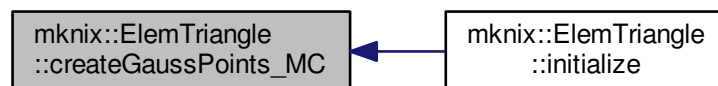
Reimplemented from [mknix::Cell](#).

Definition at line 75 of file `elemtriangle.cpp`.

6.27.3.2 `void mknix::ElemTriangle::createGaussPoints_MC ( )` [protected]

Definition at line 86 of file `elemtriangle.cpp`.

Here is the caller graph for this function:

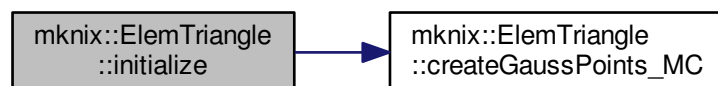


6.27.3.3 `void mknix::ElemTriangle::initialize ( std::vector< Node * > & nodes_in )` [virtual]

Reimplemented from [mknix::Cell](#).

Definition at line 55 of file `elemtriangle.cpp`.

Here is the call graph for this function:



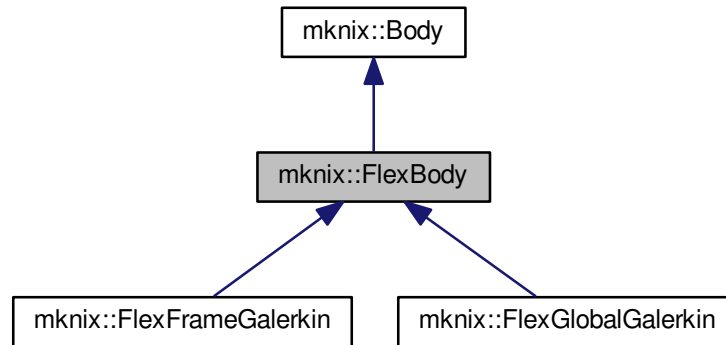
The documentation for this class was generated from the following files:

- [elemtriangle.h](#)
- [elemtriangle.cpp](#)

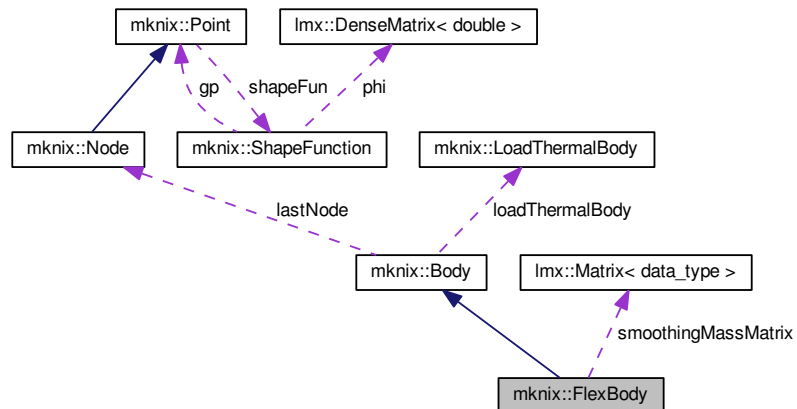
## 6.28 mknix::FlexBody Class Reference

```
#include <bodyflex.h>
```

Inheritance diagram for mknix::FlexBody:



Collaboration diagram for mknix::FlexBody:



### Public Member Functions

- `FlexBody ()`
- `FlexBody (std::string)`
- `virtual ~FlexBody ()`
- `virtual void initialize ()`

*Cascade initialization funtion. Calls the initialize methods for each of the Cells and tells them to compute their shapefunction values. Both loops are parallelized.*

- `Point * getBodyPoint (int point_number)`
- `virtual Node * getNode (int node_number)`
- `Point * getLastBodyPoint ()`

- virtual void [setType](#) (std::string type\_in)=0
- virtual void [setFormulation](#) (std::string formulation\_in)
- virtual void [calcInternalForces](#) ()=0
- virtual void [calcTangentMatrix](#) ()=0
- virtual void [assembleInternalForces](#) (Imx::Vector< [data\\_type](#) > &)=0
- virtual void [assembleTangentMatrix](#) (Imx::Matrix< [data\\_type](#) > &)=0
- void [addBodyPoint](#) ([Point](#) \*, std::string)
- void [addPoint](#) ([Node](#) \*)
- void [addPoint](#) (int, double, double, double, double, double)
- virtual int [getNumberOfPoints](#) ()
- void [setOutput](#) (std::string)  
*Activates a flag for output data at the end of the analysis.*
- void [outputToFile](#) (std::ofstream \*)  
*Streams the data stored during the analysis to a file.*
- void [writeBodyInfo](#) (std::ofstream \*)
- void [writeBoundaryNodes](#) (std::vector< [Point](#) \* > &)
- void [writeBoundaryConnectivity](#) (std::vector< std::vector< [Point](#) \* > > &)

#### Protected Attributes

- std::string [formulation](#)
- std::vector< [Node](#) \* > [points](#)
- std::vector< [Point](#) \* > [bodyPoints](#)
- bool [computeStress](#)
- bool [computeEnergy](#)
- Imx::Matrix< [data\\_type](#) > [smoothingMassMatrix](#)
- std::vector< Imx::Vector< [data\\_type](#) > > [stresses](#)
- std::vector< Imx::Vector< [data\\_type](#) > \* > [energies](#)

#### 6.28.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 34 of file bodyflex.h.

#### 6.28.2 Constructor & Destructor Documentation

##### 6.28.2.1 mknix::FlexBody::FlexBody ( )

Definition at line 27 of file bodyflex.cpp.

##### 6.28.2.2 mknix::FlexBody::FlexBody ( std::string title\_in )

Definition at line 36 of file bodyflex.cpp.

##### 6.28.2.3 mknix::FlexBody::~FlexBody ( ) [virtual]

Definition at line 45 of file bodyflex.cpp.

#### 6.28.3 Member Function Documentation

##### 6.28.3.1 void mknix::FlexBody::addBodyPoint ( [Point](#) \* point\_in, std::string method\_in )

Definition at line 65 of file bodyflex.cpp.

6.28.3.2 `void mknix::FlexBody::addPoint ( Node * node_in )`

Definition at line 79 of file `bodyflex.cpp`.

6.28.3.3 `void mknix::FlexBody::addPoint ( int nodeNumber, double x, double y, double z, double alpha, double dc )`

Definition at line 89 of file `bodyflex.cpp`.

6.28.3.4 `virtual void mknix::FlexBody::assembleInternalForces ( Imx::Vector< data_type > & ) [pure virtual]`

Implemented in [mknix::FlexFrameGalerkin](#), and [mknix::FlexGlobalGalerkin](#).

6.28.3.5 `virtual void mknix::FlexBody::assembleTangentMatrix ( Imx::Matrix< data_type > & ) [pure virtual]`

Implemented in [mknix::FlexFrameGalerkin](#), and [mknix::FlexGlobalGalerkin](#).

6.28.3.6 `virtual void mknix::FlexBody::calcInternalForces ( ) [pure virtual]`

Implemented in [mknix::FlexFrameGalerkin](#), and [mknix::FlexGlobalGalerkin](#).

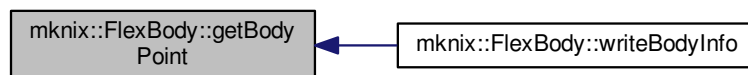
6.28.3.7 `virtual void mknix::FlexBody::calcTangentMatrix ( ) [pure virtual]`

Implemented in [mknix::FlexFrameGalerkin](#), and [mknix::FlexGlobalGalerkin](#).

6.28.3.8 `Point* mknix::FlexBody::getBodyPoint ( int point_number ) [inline]`

Definition at line 45 of file `bodyflex.h`.

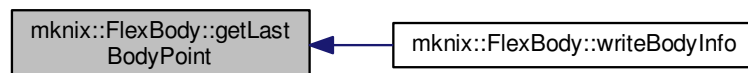
Here is the caller graph for this function:



6.28.3.9 `Point* mknix::FlexBody::getLastBodyPoint ( ) [inline]`

Definition at line 56 of file `bodyflex.h`.

Here is the caller graph for this function:



6.28.3.10 `virtual Node* mknix::FlexBody::getNode ( int node_number ) [inline],[virtual]`

Reimplemented from [mknix::Body](#).



Definition at line 51 of file bodyflex.h.

Here is the caller graph for this function:



#### 6.28.3.11 `virtual int mknix::FlexBody::getNumberOfPoints ( ) [inline],[virtual]`

Definition at line 85 of file bodyflex.h.

#### 6.28.3.12 `void mknix::FlexBody::initialize ( ) [virtual]`

Cascade initialization funtion. Calls the initialize methods for each of the Cells and tells them to compute their shapefunction values. Both loops are parallelized.

##### Returns

void

Reimplemented from [mknix::Body](#).

Definition at line 49 of file bodyflex.cpp.

Here is the call graph for this function:



#### 6.28.3.13 `void mknix::FlexBody::outputToFile ( std::ofstream * outFile ) [virtual]`

Streams the data stored during the analysis to a file.

##### Parameters

<i>outFile</i>	Output files
----------------	--------------

##### Returns

void

Reimplemented from [mknix::Body](#).

Definition at line 128 of file bodyflex.cpp.

Here is the call graph for this function:



**6.28.3.14** `virtual void mknix::FlexBody::setFormulation ( std::string formulation_in )` `[inline],[virtual]`

Definition at line 63 of file bodyflex.h.

**6.28.3.15** `void mknix::FlexBody::setOutput ( std::string outputType_in )` `[virtual]`

Activates a flag for output data at the end of the analysis.

See also

[outputToFile\(\)](#)  
[outputStep\(\)](#)

Parameters

<i>outputType_in</i>	Keyword of the flag. Options are: [STRESS, ENERGY]
----------------------	--

Returns

void

Implements [mknix::Body](#).

Definition at line 112 of file bodyflex.cpp.

**6.28.3.16** `virtual void mknix::FlexBody::setType ( std::string type_in )` `[pure virtual]`

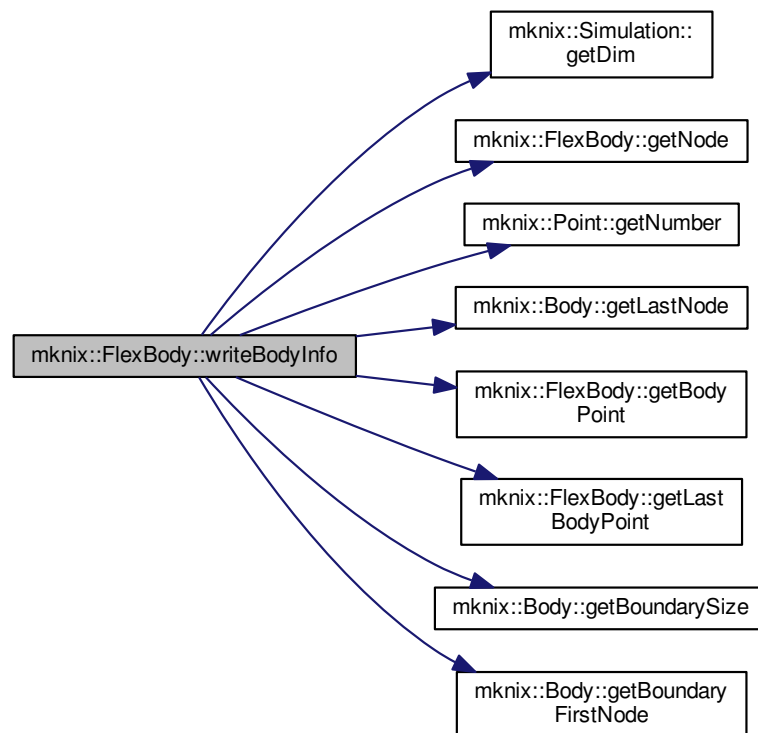
Implemented in [mknix::FlexFrameGalerkin](#), and [mknix::FlexGlobalGalerkin](#).

**6.28.3.17** `void mknix::FlexBody::writeBodyInfo ( std::ofstream * outFile )` `[virtual]`

Implements [mknix::Body](#).

Definition at line 155 of file bodyflex.cpp.

Here is the call graph for this function:



**6.28.3.18** `void mknix::FlexBody::writeBoundaryConnectivity ( std::vector< std::vector< Point * > > & connectivity_nodes )` [virtual]

Implements [mknix::Body](#).

Definition at line 240 of file `bodyflex.cpp`.

**6.28.3.19** `void mknix::FlexBody::writeBoundaryNodes ( std::vector< Point * > & boundary_nodes )` [virtual]

Implements [mknix::Body](#).

Definition at line 233 of file `bodyflex.cpp`.

## 6.28.4 Member Data Documentation

**6.28.4.1** `std::vector<Point*> mknix::FlexBody::bodyPoints` [protected]

Points to define integration domain

Definition at line 107 of file `bodyflex.h`.

**6.28.4.2** `bool mknix::FlexBody::computeEnergy` [protected]

Definition at line 109 of file `bodyflex.h`.

#### 6.28.4.3 `bool mknix::FlexBody::computeStress` [protected]

Definition at line 108 of file `bodyflex.h`.

#### 6.28.4.4 `std::vector<Imx::Vector<data_type>*> mknix::FlexBody::energies` [protected]

Definition at line 113 of file `bodyflex.h`.

#### 6.28.4.5 `std::string mknix::FlexBody::formulation` [protected]

Definition at line 105 of file `bodyflex.h`.

#### 6.28.4.6 `std::vector<Node*> mknix::FlexBody::points` [protected]

Additional points to define loads or constraints

Definition at line 106 of file `bodyflex.h`.

#### 6.28.4.7 `Imx::Matrix<data_type> mknix::FlexBody::smoothingMassMatrix` [protected]

Definition at line 111 of file `bodyflex.h`.

#### 6.28.4.8 `std::vector<Imx::Vector<data_type>> mknix::FlexBody::stresses` [protected]

Definition at line 112 of file `bodyflex.h`.

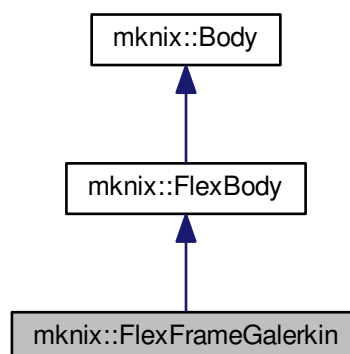
The documentation for this class was generated from the following files:

- [bodyflex.h](#)
- [bodyflex.cpp](#)

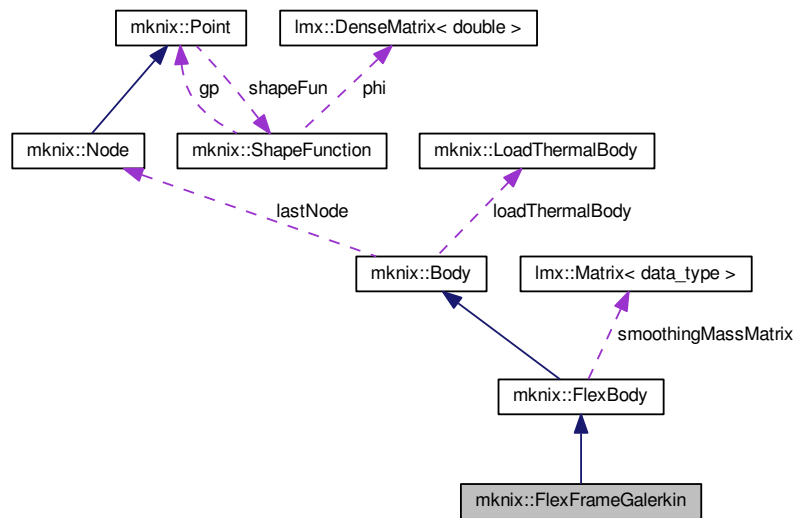
## 6.29 mknix::FlexFrameGalerkin Class Reference

```
#include <bodyflexframegalerkin.h>
```

Inheritance diagram for `mknix::FlexFrameGalerkin`:



Collaboration diagram for mknix::FlexFrameGalerkin:



### Public Member Functions

- [FlexFrameGalerkin](#) ()
  - [FlexFrameGalerkin](#) (std::string)
  - [~FlexFrameGalerkin](#) ()
  - std::string [getType](#) ()
  - void [setType](#) (std::string type\_in)
  - void [calcMassMatrix](#) ()
  - void [calcInternalForces](#) ()
  - void [calcExternalForces](#) ()
  - void [calcTangentMatrix](#) ()
  - void [assembleMassMatrix](#) (Imx::Matrix< data\_type > &)
  - void [assembleInternalForces](#) (Imx::Vector< data\_type > &)
  - void [assembleExternalForces](#) (Imx::Vector< data\_type > &)
  - void [assembleTangentMatrix](#) (Imx::Matrix< data\_type > &)
  - void [outputStep](#) (const Imx::Vector< data\_type > &, const Imx::Vector< data\_type > &)
- Postprocess and store step results for dynamic analysis.*
- void [outputStep](#) (const Imx::Vector< data\_type > &)
- Postprocess and store step results for static analysis.*

### Additional Inherited Members

#### 6.29.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 30 of file `bodyflexframegalerkin.h`.

## 6.29.2 Constructor & Destructor Documentation

### 6.29.2.1 `mknix::FlexFrameGalerkin::FlexFrameGalerkin ( )`

Definition at line 27 of file `bodyflexframegalerkin.cpp`.

### 6.29.2.2 `mknix::FlexFrameGalerkin::FlexFrameGalerkin ( std::string title_in )`

Definition at line 33 of file `bodyflexframegalerkin.cpp`.

### 6.29.2.3 `mknix::FlexFrameGalerkin::~~FlexFrameGalerkin ( )`

Definition at line 39 of file `bodyflexframegalerkin.cpp`.

## 6.29.3 Member Function Documentation

### 6.29.3.1 `void mknix::FlexFrameGalerkin::assembleExternalForces ( Imx::Vector< data_type > & globalExternalForces )` [virtual]

Implements [mknix::Body](#).

Definition at line 151 of file `bodyflexframegalerkin.cpp`.

### 6.29.3.2 `void mknix::FlexFrameGalerkin::assembleInternalForces ( Imx::Vector< data_type > & globalInternalForces )` [virtual]

Implements [mknix::FlexBody](#).

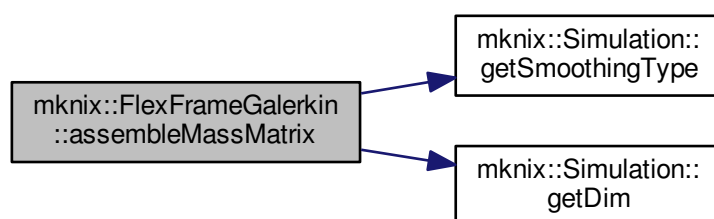
Definition at line 139 of file `bodyflexframegalerkin.cpp`.

### 6.29.3.3 `void mknix::FlexFrameGalerkin::assembleMassMatrix ( Imx::Matrix< data_type > & globalMass )` [virtual]

Implements [mknix::Body](#).

Definition at line 112 of file `bodyflexframegalerkin.cpp`.

Here is the call graph for this function:



### 6.29.3.4 `void mknix::FlexFrameGalerkin::assembleTangentMatrix ( Imx::Matrix< data_type > & globalTangent )` [virtual]

Implements [mknix::FlexBody](#).

Definition at line 162 of file `bodyflexframegalerkin.cpp`.

6.29.3.5 `void mknix::FlexFrameGalerkin::calcExternalForces ( ) [virtual]`

Implements [mknix::Body](#).

Definition at line 77 of file `bodyflexframegalerkin.cpp`.

6.29.3.6 `void mknix::FlexFrameGalerkin::calcInternalForces ( ) [virtual]`

Implements [mknix::FlexBody](#).

Definition at line 55 of file `bodyflexframegalerkin.cpp`.

6.29.3.7 `void mknix::FlexFrameGalerkin::calcMassMatrix ( ) [virtual]`

Implements [mknix::Body](#).

Definition at line 43 of file `bodyflexframegalerkin.cpp`.

6.29.3.8 `void mknix::FlexFrameGalerkin::calcTangentMatrix ( ) [virtual]`

Implements [mknix::FlexBody](#).

Definition at line 89 of file `bodyflexframegalerkin.cpp`.

6.29.3.9 `std::string mknix::FlexFrameGalerkin::getType ( ) [inline],[virtual]`

Implements [mknix::Body](#).

Definition at line 39 of file `bodyflexframegalerkin.h`.

6.29.3.10 `void mknix::FlexFrameGalerkin::outputStep ( const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot ) [virtual]`

Postprocess and store step results for dynamic analysis.

Parameters

<i>q</i>	Global configuration vector
<i>qdot</i>	Global configuration first derivative vector

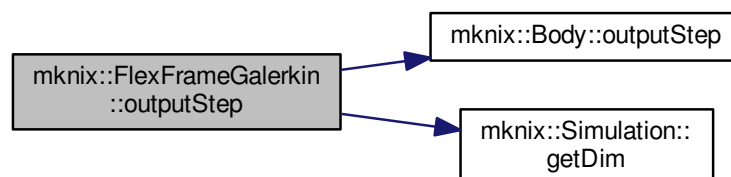
Returns

void

Implements [mknix::Body](#).

Definition at line 180 of file `bodyflexframegalerkin.cpp`.

Here is the call graph for this function:



6.29.3.11 `void mknix::FlexFrameGalerkin::outputStep ( const Imx::Vector< data_type > & q )` [virtual]

Postprocess and store step results for static analysis.



## Parameters

$q$	Global configuration vector
-----	-----------------------------

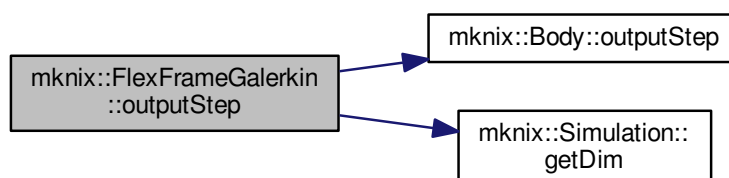
## Returns

void

Implements [mknix::Body](#).

Definition at line 237 of file bodyflexframegalerkin.cpp.

Here is the call graph for this function:



**6.29.3.12** `void mknix::FlexFrameGalerkin::setType ( std::string type_in )` `[inline], [virtual]`

Implements [mknix::FlexBody](#).

Definition at line 43 of file bodyflexframegalerkin.h.

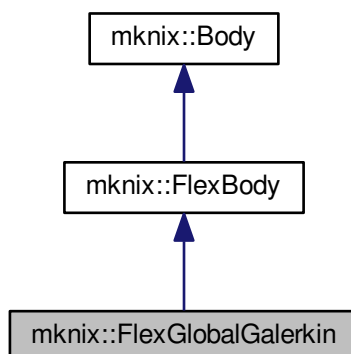
The documentation for this class was generated from the following files:

- [bodyflexframegalerkin.h](#)
- [bodyflexframegalerkin.cpp](#)

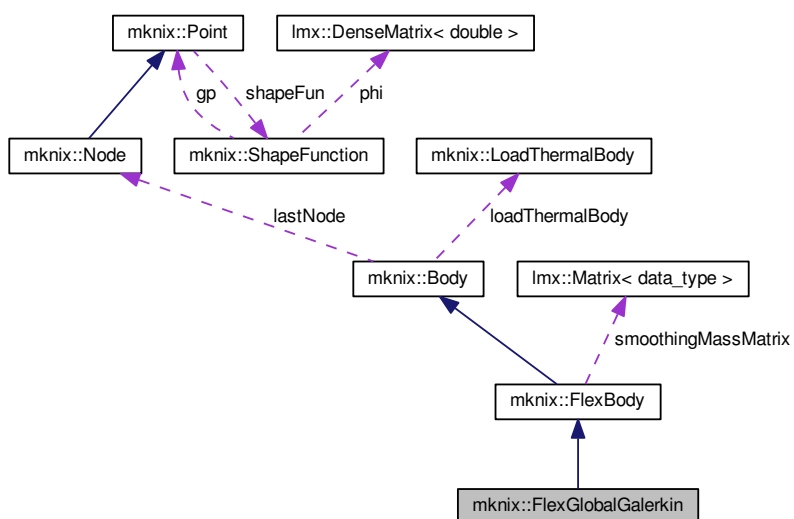
## 6.30 mknix::FlexGlobalGalerkin Class Reference

```
#include <bodyflexglobalgalerkin.h>
```

Inheritance diagram for `mknix::FlexGlobalGalerkin`:



Collaboration diagram for `mknix::FlexGlobalGalerkin`:



#### Public Member Functions

- [FlexGlobalGalerkin](#) ()
- [FlexGlobalGalerkin](#) (std::string)
- [~FlexGlobalGalerkin](#) ()
- std::string [getType](#) ()
- void [setType](#) (std::string type\_in)
- void [calcMassMatrix](#) ()
- void [calcInternalForces](#) ()
- void [calcExternalForces](#) ()

- void [calcTangentMatrix](#) ()
- void [assembleMassMatrix](#) (Imx::Matrix< data\_type > &)
- void [assembleInternalForces](#) (Imx::Vector< data\_type > &)
- void [assembleExternalForces](#) (Imx::Vector< data\_type > &)
- void [assembleTangentMatrix](#) (Imx::Matrix< data\_type > &)
- void [outputStep](#) (const Imx::Vector< data\_type > &, const Imx::Vector< data\_type > &)  
*Postprocess and store step results for dynamic analysis.*
- void [outputStep](#) (const Imx::Vector< data\_type > &)  
*Postprocess and store step results for static analysis.*

## Additional Inherited Members

### 6.30.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 30 of file bodyflexglobalgalerkin.h.

### 6.30.2 Constructor & Destructor Documentation

#### 6.30.2.1 mknix::FlexGlobalGalerkin::FlexGlobalGalerkin ( )

Definition at line 27 of file bodyflexglobalgalerkin.cpp.

#### 6.30.2.2 mknix::FlexGlobalGalerkin::FlexGlobalGalerkin ( std::string title\_in )

Definition at line 33 of file bodyflexglobalgalerkin.cpp.

#### 6.30.2.3 mknix::FlexGlobalGalerkin::~~FlexGlobalGalerkin ( )

Definition at line 39 of file bodyflexglobalgalerkin.cpp.

### 6.30.3 Member Function Documentation

#### 6.30.3.1 void mknix::FlexGlobalGalerkin::assembleExternalForces ( Imx::Vector< data\_type > & globalExternalForces ) [virtual]

Implements [mknix::Body](#).

Definition at line 136 of file bodyflexglobalgalerkin.cpp.

#### 6.30.3.2 void mknix::FlexGlobalGalerkin::assembleInternalForces ( Imx::Vector< data\_type > & globalInternalForces ) [virtual]

Implements [mknix::FlexBody](#).

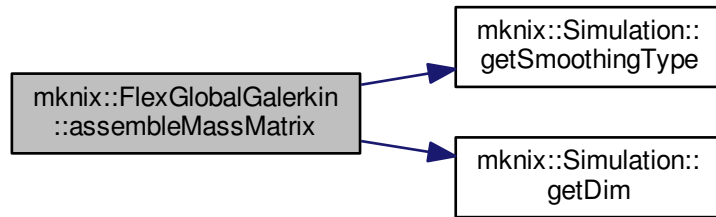
Definition at line 126 of file bodyflexglobalgalerkin.cpp.

#### 6.30.3.3 void mknix::FlexGlobalGalerkin::assembleMassMatrix ( Imx::Matrix< data\_type > & globalMass ) [virtual]

Implements [mknix::Body](#).

Definition at line 101 of file bodyflexglobalgalerkin.cpp.

Here is the call graph for this function:



6.30.3.4 `void mknix::FlexGlobalGalerkin::assembleTangentMatrix ( Imx::Matrix< data_type > & globalTangent )`  
[virtual]

Implements [mknix::FlexBody](#).

Definition at line 145 of file `bodyflexglobalgalerkin.cpp`.

6.30.3.5 `void mknix::FlexGlobalGalerkin::calcExternalForces ( )` [virtual]

Implements [mknix::Body](#).

Definition at line 72 of file `bodyflexglobalgalerkin.cpp`.

6.30.3.6 `void mknix::FlexGlobalGalerkin::calcInternalForces ( )` [virtual]

Implements [mknix::FlexBody](#).

Definition at line 54 of file `bodyflexglobalgalerkin.cpp`.

6.30.3.7 `void mknix::FlexGlobalGalerkin::calcMassMatrix ( )` [virtual]

Implements [mknix::Body](#).

Definition at line 43 of file `bodyflexglobalgalerkin.cpp`.

6.30.3.8 `void mknix::FlexGlobalGalerkin::calcTangentMatrix ( )` [virtual]

Implements [mknix::FlexBody](#).

Definition at line 82 of file `bodyflexglobalgalerkin.cpp`.

6.30.3.9 `std::string mknix::FlexGlobalGalerkin::getType ( )` [inline], [virtual]

Implements [mknix::Body](#).

Definition at line 39 of file `bodyflexglobalgalerkin.h`.

6.30.3.10 `void mknix::FlexGlobalGalerkin::outputStep ( const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot )` [virtual]

Postprocess and store step results for dynamic analysis.

## Parameters

$q$	Global configuration vector
$\dot{q}$	Global configuration first derivative vector

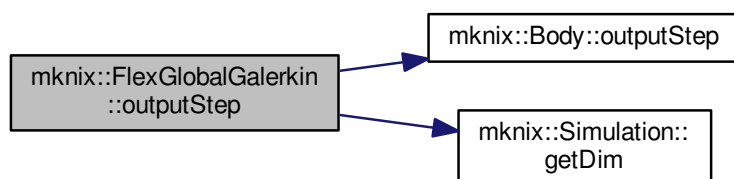
## Returns

void

Implements [mknix::Body](#).

Definition at line 161 of file bodyflexglobalgalerkin.cpp.

Here is the call graph for this function:



## 6.30.3.11 void mknix::FlexGlobalGalerkin::outputStep ( const Imx::Vector&lt; data\_type &gt; &amp; q ) [virtual]

Postprocess and store step results for static analysis.

## Parameters

$q$	Global configuration vector
-----	-----------------------------

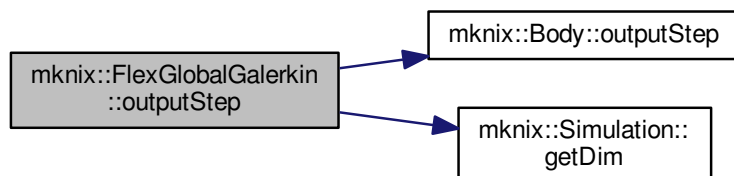
## Returns

void

Implements [mknix::Body](#).

Definition at line 212 of file bodyflexglobalgalerkin.cpp.

Here is the call graph for this function:



6.30.3.12 `void mknix::FlexGlobalGalerkin::setType ( std::string type_in ) [inline],[virtual]`

Implements [mknix::FlexBody](#).

Definition at line 43 of file `bodyflexglobalgalerkin.h`.

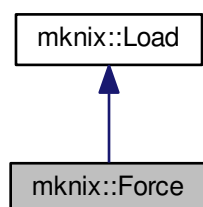
The documentation for this class was generated from the following files:

- [bodyflexglobalgalerkin.h](#)
- [bodyflexglobalgalerkin.cpp](#)

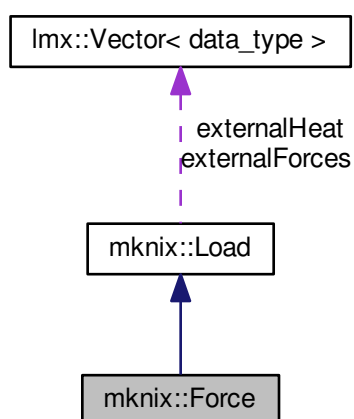
## 6.31 mknix::Force Class Reference

```
#include <force.h>
```

Inheritance diagram for `mknix::Force`:



Collaboration diagram for `mknix::Force`:



### Public Member Functions

- [Force](#) ()

- [Force](#) ([Node](#) \*, double, double, double)
- [~Force](#) ()
- virtual void [outputToFile](#) (std::ofstream \*)

#### Additional Inherited Members

##### 6.31.1 Detailed Description

###### Author

AUTHORS <MAILS>

Definition at line 30 of file force.h.

##### 6.31.2 Constructor & Destructor Documentation

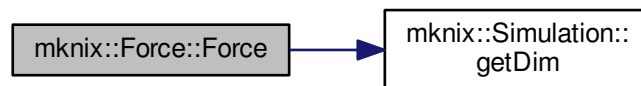
###### 6.31.2.1 mknix::Force::Force ( )

Definition at line 27 of file force.cpp.

###### 6.31.2.2 mknix::Force::Force ( [Node](#) \* *node\_in*, double *fx\_in*, double *fy\_in*, double *fz\_in* )

Definition at line 32 of file force.cpp.

Here is the call graph for this function:



###### 6.31.2.3 mknix::Force::~~Force ( )

Definition at line 44 of file force.cpp.

##### 6.31.3 Member Function Documentation

###### 6.31.3.1 void mknix::Force::outputToFile ( std::ofstream \* *outFile* ) [virtual]

Implements [mknix::Load](#).

Definition at line 48 of file force.cpp.

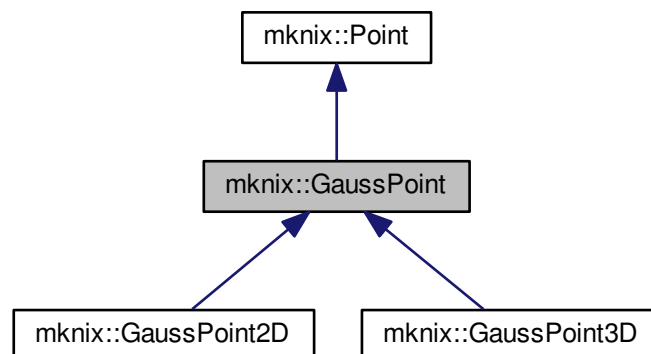
The documentation for this class was generated from the following files:

- [force.h](#)
- [force.cpp](#)

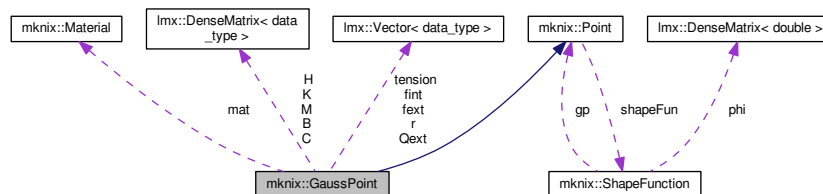
## 6.32 mknix::GaussPoint Class Reference

```
#include <gausspoint.h>
```

Inheritance diagram for `mknix::GaussPoint`:



Collaboration diagram for `mknix::GaussPoint`:



## Public Member Functions

- [GaussPoint](#) ()
- [GaussPoint](#) (int dim\_in, double alpha\_in, double weight\_in, double jacobian\_in, [Material](#) \*mat\_in, int num, double coor\_x, double coor\_y, double dc\_in, bool)
- [GaussPoint](#) (int dim\_in, double alpha\_in, double weight\_in, double jacobian\_in, [Material](#) \*mat\_in, int num, double coor\_x, double coor\_y, double coor\_z, double dc\_in, bool)
- virtual [~GaussPoint](#) ()
- virtual void [shapeFunSolve](#) (std::string, double) override
- virtual void [fillFEmatrices](#) ()=0
- void [computeCij](#) ()
- void [computeHij](#) ()
- void [computeQext](#) ([LoadThermalBody](#) \*)
- virtual void [computeFint](#) ()=0
- virtual void [computeFext](#) ()=0
- virtual void [computeMij](#) ()=0
- virtual void [computeKij](#) ()=0
- virtual void [computeStress](#) ()=0
- virtual void [computeNLStress](#) ()=0
- virtual void [computeNLFint](#) ()=0
- virtual void [computeNLKij](#) ()=0



- void `assembleCij` (`Imx::Matrix< data_type > &`)
- void `assembleHij` (`Imx::Matrix< data_type > &`)
- void `assembleQext` (`Imx::Vector< data_type > &`)
- virtual void `assembleMij` (`Imx::Matrix< data_type > &`)=0
- virtual void `assembleKij` (`Imx::Matrix< data_type > &`)=0
- virtual void `assembleRi` (`Imx::Vector< data_type > &`, `int`)=0
- virtual void `assembleFint` (`Imx::Vector< data_type > &`)=0
- virtual void `assembleFext` (`Imx::Vector< data_type > &`)=0
- virtual double `calcPotentialE` (`const Imx::Vector< data_type > &`)=0
- virtual double `calcKineticE` (`const Imx::Vector< data_type > &`)=0
- virtual double `calcElasticE` (`()`)=0
- void `gnuplotOutStress` (`std::ofstream &`)

#### Protected Attributes

- int `num`
- double `weight`
- `Material * mat`
- bool `stressPoint`
- `Imx::DenseMatrix< data_type > B`
- `Imx::DenseMatrix< data_type > C`
- `Imx::DenseMatrix< data_type > H`
- `Imx::DenseMatrix< data_type > M`
- `Imx::DenseMatrix< data_type > K`
- `Imx::Vector< data_type > tension`
- `Imx::Vector< data_type > r`
- `Imx::Vector< data_type > Qext`
- `Imx::Vector< data_type > fint`
- `Imx::Vector< data_type > fext`

#### 6.32.1 Detailed Description

##### Author

Daniel Iglesias

Definition at line 45 of file `gausspoint.h`.

#### 6.32.2 Constructor & Destructor Documentation

##### 6.32.2.1 `mknix::GaussPoint::GaussPoint ( )`

Definition at line 13 of file `gausspoint.cpp`.

##### 6.32.2.2 `mknix::GaussPoint::GaussPoint ( int dim_in, double alpha_in, double weight_in, double jacobian_in, Material * mat_in, int num, double coor_x, double coor_y, double dc_in, bool stressPoint_in )`

Definition at line 18 of file `gausspoint.cpp`.

##### 6.32.2.3 `mknix::GaussPoint::GaussPoint ( int dim_in, double alpha_in, double weight_in, double jacobian_in, Material * mat_in, int num, double coor_x, double coor_y, double coor_z, double dc_in, bool stressPoint_in )`

Definition at line 36 of file `gausspoint.cpp`.

#### 6.32.2.4 `mknix::GaussPoint::~~GaussPoint ( ) [virtual]`

Definition at line 55 of file `gausspoint.cpp`.

### 6.32.3 Member Function Documentation

#### 6.32.3.1 `void mknix::GaussPoint::assembleCij ( Imx::Matrix< data_type > & globalCapacity )`

Definition at line 190 of file `gausspoint.cpp`.

#### 6.32.3.2 `virtual void mknix::GaussPoint::assembleFext ( Imx::Vector< data_type > & ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

#### 6.32.3.3 `virtual void mknix::GaussPoint::assembleFint ( Imx::Vector< data_type > & ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

#### 6.32.3.4 `void mknix::GaussPoint::assembleHij ( Imx::Matrix< data_type > & globalConductivity )`

Definition at line 202 of file `gausspoint.cpp`.

#### 6.32.3.5 `virtual void mknix::GaussPoint::assembleKij ( Imx::Matrix< data_type > & ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

#### 6.32.3.6 `virtual void mknix::GaussPoint::assembleMij ( Imx::Matrix< data_type > & ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

#### 6.32.3.7 `void mknix::GaussPoint::assembleQext ( Imx::Vector< data_type > & globalHeat )`

Definition at line 214 of file `gausspoint.cpp`.

#### 6.32.3.8 `virtual void mknix::GaussPoint::assembleRi ( Imx::Vector< data_type > &, int ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

#### 6.32.3.9 `virtual double mknix::GaussPoint::calcElasticE ( ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

#### 6.32.3.10 `virtual double mknix::GaussPoint::calcKineticE ( const Imx::Vector< data_type > & ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

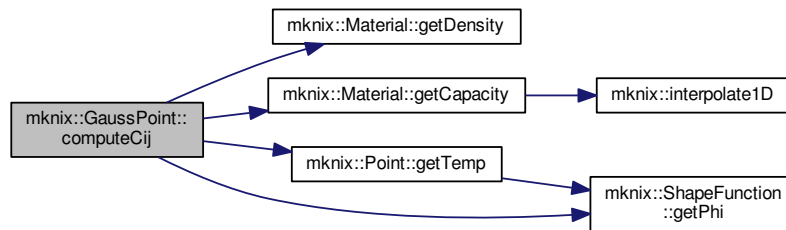
#### 6.32.3.11 `virtual double mknix::GaussPoint::calcPotentialE ( const Imx::Vector< data_type > & ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

#### 6.32.3.12 `void mknix::GaussPoint::computeCij ( )`

Definition at line 96 of file `gausspoint.cpp`.

Here is the call graph for this function:



6.32.3.13 `virtual void mknix::GaussPoint::computeFext ( )` [pure virtual]

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

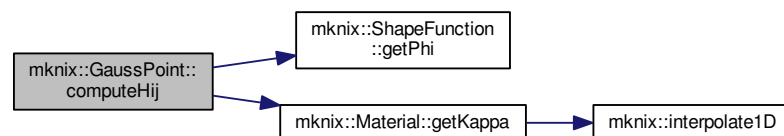
6.32.3.14 `virtual void mknix::GaussPoint::computeFint ( )` [pure virtual]

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

6.32.3.15 `void mknix::GaussPoint::computeHij ( )`

Definition at line 128 of file `gausspoint.cpp`.

Here is the call graph for this function:



6.32.3.16 `virtual void mknix::GaussPoint::computeKij ( )` [pure virtual]

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

6.32.3.17 `virtual void mknix::GaussPoint::computeMij ( )` [pure virtual]

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

6.32.3.18 `virtual void mknix::GaussPoint::computeNLFint ( )` [pure virtual]

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

6.32.3.19 `virtual void mknix::GaussPoint::computeNLKij ( )` [pure virtual]

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

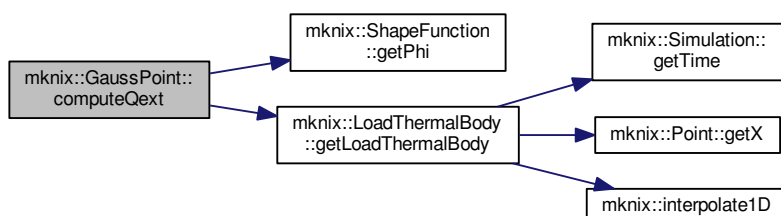
6.32.3.20 `virtual void mknix::GaussPoint::computeNLStress ( )` [pure virtual]

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

6.32.3.21 `void mknix::GaussPoint::computeQext ( LoadThermalBody * loadThermalBody_in )`

Definition at line 179 of file gausspoint.cpp.

Here is the call graph for this function:



6.32.3.22 `virtual void mknix::GaussPoint::computeStress ( ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

6.32.3.23 `virtual void mknix::GaussPoint::fillFEmatrices ( ) [pure virtual]`

Implemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

6.32.3.24 `void mknix::GaussPoint::gnuplotOutStress ( std::ofstream & gptension )`

Definition at line 224 of file gausspoint.cpp.

6.32.3.25 `void mknix::GaussPoint::shapeFunSolve ( std::string type_in, double q_in ) [override],[virtual]`

Reimplemented from [mknix::Point](#).

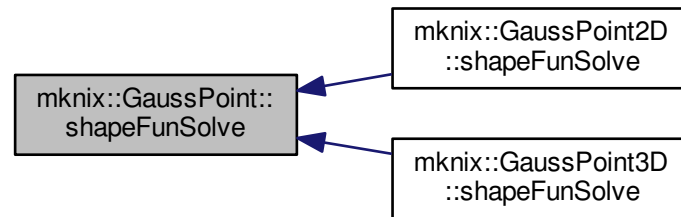
Reimplemented in [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

Definition at line 60 of file gausspoint.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.32.4 Member Data Documentation

**6.32.4.1** `Imx::DenseMatrix<data_type> mknix::GaussPoint::B` [protected]

Definition at line 116 of file gausspoint.h.

**6.32.4.2** `Imx::DenseMatrix<data_type> mknix::GaussPoint::C` [protected]

Definition at line 117 of file gausspoint.h.

**6.32.4.3** `Imx::Vector<data_type> mknix::GaussPoint::fext` [protected]

Definition at line 127 of file gausspoint.h.

**6.32.4.4** `Imx::Vector<data_type> mknix::GaussPoint::fint` [protected]

Definition at line 126 of file gausspoint.h.

**6.32.4.5** `Imx::DenseMatrix<data_type> mknix::GaussPoint::H` [protected]

Definition at line 118 of file gausspoint.h.

**6.32.4.6** `Imx::DenseMatrix<data_type> mknix::GaussPoint::K` [protected]

Definition at line 120 of file gausspoint.h.

**6.32.4.7** `Imx::DenseMatrix<data_type> mknix::GaussPoint::M` [protected]

Definition at line 119 of file gausspoint.h.

**6.32.4.8** `Material* mknix::GaussPoint::mat` [protected]

Definition at line 113 of file gausspoint.h.

**6.32.4.9** `int mknix::GaussPoint::num` [protected]

Definition at line 111 of file gausspoint.h.

**6.32.4.10** `Imx::Vector<data_type> mknix::GaussPoint::Qext` [protected]

Definition at line 125 of file gausspoint.h.

#### 6.32.4.11 `Imx::Vector<data_type> mknix::GaussPoint::r` [protected]

Definition at line 123 of file gausspoint.h.

#### 6.32.4.12 `bool mknix::GaussPoint::stressPoint` [protected]

Definition at line 114 of file gausspoint.h.

#### 6.32.4.13 `Imx::Vector<data_type> mknix::GaussPoint::tension` [protected]

Definition at line 122 of file gausspoint.h.

#### 6.32.4.14 `double mknix::GaussPoint::weight` [protected]

Definition at line 112 of file gausspoint.h.

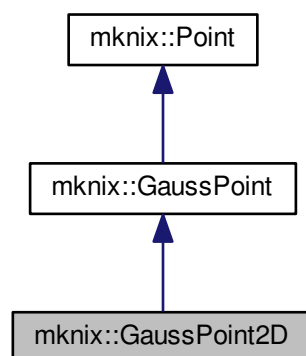
The documentation for this class was generated from the following files:

- [gausspoint.h](#)
- [gausspoint.cpp](#)

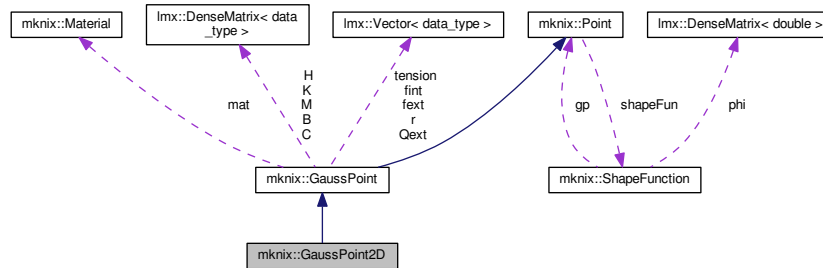
### 6.33 `mknix::GaussPoint2D` Class Reference

```
#include <gausspoint2D.h>
```

Inheritance diagram for `mknix::GaussPoint2D`:



Collaboration diagram for mknix::GaussPoint2D:



### Public Member Functions

- [GaussPoint2D](#) ()
- [GaussPoint2D](#) (double alpha\_in, double weight\_in, double jacobian\_in, [Material](#) \*mat\_in, int num, double coor\_x, double coor\_y, double dc\_in, bool stressPoint\_in)
- [GaussPoint2D](#) (double alpha\_in, double weight\_in, double jacobian\_in, [Material](#) \*mat\_in, int num, double coor\_x, double coor\_y, double coor\_z, double dc\_in, bool stressPoint\_in)
- [~GaussPoint2D](#) ()
- void [shapeFunSolve](#) (std::string, double) override
- void [fillFEmatrices](#) () override
- void [computeMij](#) () override
- void [computeKij](#) () override
- void [computeStress](#) () override
- void [computeNLStress](#) () override
- void [computeFint](#) () override
- void [computeFext](#) () override
- void [computeNLFint](#) () override
- void [computeNLKij](#) () override
- void [assembleMij](#) (Imx::Matrix< data\_type > &) override
- void [assembleKij](#) (Imx::Matrix< data\_type > &) override
- void [assembleRi](#) (Imx::Vector< data\_type > &, int) override
- void [assembleFint](#) (Imx::Vector< data\_type > &) override
- void [assembleFext](#) (Imx::Vector< data\_type > &) override
- double [calcPotentialE](#) (const Imx::Vector< data\_type > &) override
- double [calcKineticE](#) (const Imx::Vector< data\_type > &) override
- double [calcElasticE](#) () override

### Additional Inherited Members

#### 6.33.1 Detailed Description

#### Author

Daniel Iglesias

Definition at line 45 of file gausspoint2D.h.

### 6.33.2 Constructor & Destructor Documentation

#### 6.33.2.1 `mknix::GaussPoint2D::GaussPoint2D ( )`

Definition at line 13 of file `gausspoint2D.cpp`.

#### 6.33.2.2 `mknix::GaussPoint2D::GaussPoint2D ( double alpha_in, double weight_in, double jacobian_in, Material * mat_in, int num, double coor_x, double coor_y, double dc_in, bool stressPoint_in )`

Definition at line 18 of file `gausspoint2D.cpp`.

#### 6.33.2.3 `mknix::GaussPoint2D::GaussPoint2D ( double alpha_in, double weight_in, double jacobian_in, Material * mat_in, int num, double coor_x, double coor_y, double coor_z, double dc_in, bool stressPoint_in )`

Definition at line 29 of file `gausspoint2D.cpp`.

#### 6.33.2.4 `mknix::GaussPoint2D::~~GaussPoint2D ( )`

Definition at line 40 of file `gausspoint2D.cpp`.

### 6.33.3 Member Function Documentation

#### 6.33.3.1 `void mknix::GaussPoint2D::assembleFext ( Imx::Vector< data_type > & globalFext ) [override], [virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 552 of file `gausspoint2D.cpp`.

#### 6.33.3.2 `void mknix::GaussPoint2D::assembleFint ( Imx::Vector< data_type > & globalFint ) [override], [virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 540 of file `gausspoint2D.cpp`.

#### 6.33.3.3 `void mknix::GaussPoint2D::assembleKij ( Imx::Matrix< data_type > & globalTangent ) [override], [virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 492 of file `gausspoint2D.cpp`.

#### 6.33.3.4 `void mknix::GaussPoint2D::assembleMij ( Imx::Matrix< data_type > & globalMass ) [override], [virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 471 of file `gausspoint2D.cpp`.

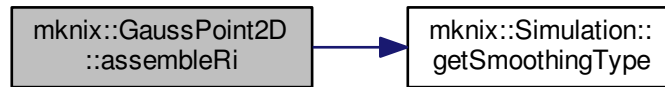
#### 6.33.3.5 `void mknix::GaussPoint2D::assembleRi ( Imx::Vector< data_type > & bodyR, int firstNode ) [override], [virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 512 of file `gausspoint2D.cpp`.



Here is the call graph for this function:

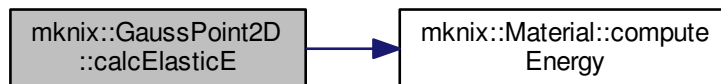


**6.33.3.6** `double mknix::GaussPoint2D::calcElasticE ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 596 of file `gausspoint2D.cpp`.

Here is the call graph for this function:



**6.33.3.7** `double mknix::GaussPoint2D::calcKineticE ( const Imx::Vector< data_type > & qdot ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 577 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

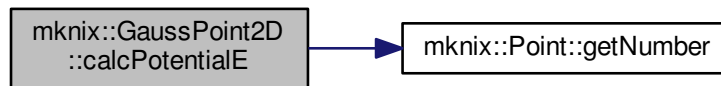


**6.33.3.8** `double mknix::GaussPoint2D::calcPotentialE ( const Imx::Vector< data_type > & q ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 564 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

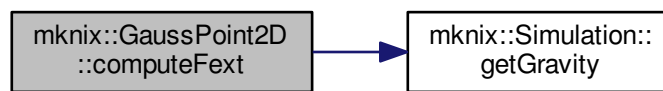


**6.33.3.9** `void mknix::GaussPoint2D::computeFext ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 271 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

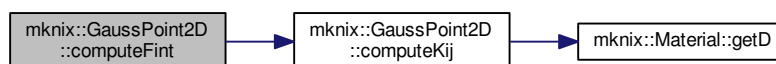


**6.33.3.10** `void mknix::GaussPoint2D::computeFint ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 255 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

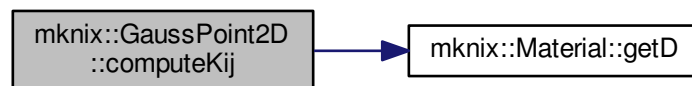


**6.33.3.11** `void mknix::GaussPoint2D::computeKij ( ) [override],[virtual]`

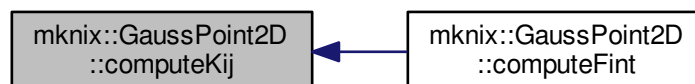
Implements [mknix::GaussPoint](#).

Definition at line 133 of file `gausspoint2D.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:

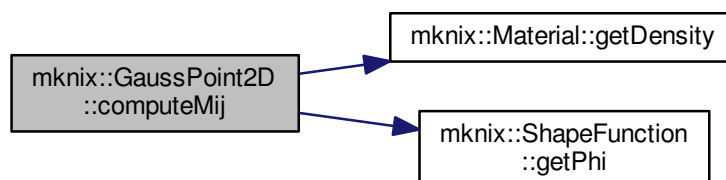


**6.33.3.12** `void mknix::GaussPoint2D::computeMij ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 113 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

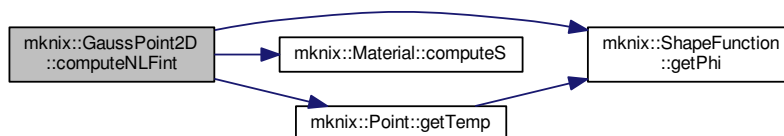


**6.33.3.13** `void mknix::GaussPoint2D::computeNLFint ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 288 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

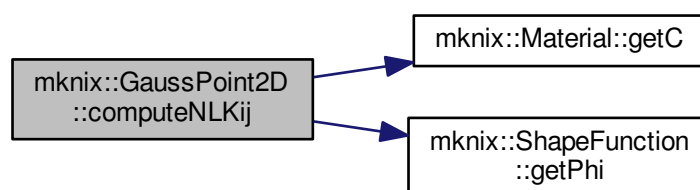


6.33.3.14 `void mknix::GaussPoint2D::computeNLKij ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 312 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

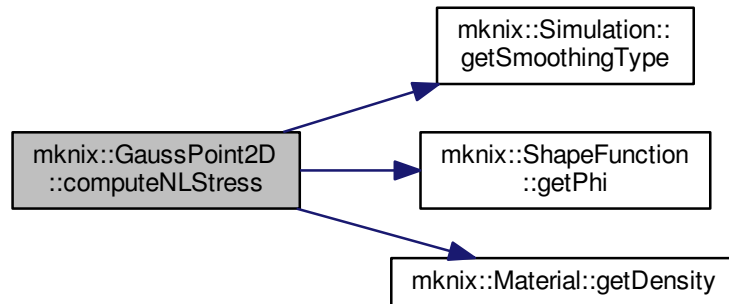


6.33.3.15 `void mknix::GaussPoint2D::computeNLStress ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 216 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

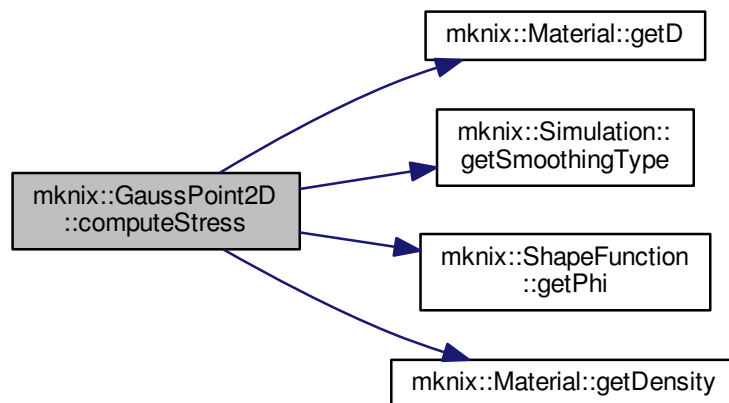


6.33.3.16 `void mknix::GaussPoint2D::computeStress ( )` [override],[virtual]

Implements [mknix::GaussPoint](#).

Definition at line 160 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

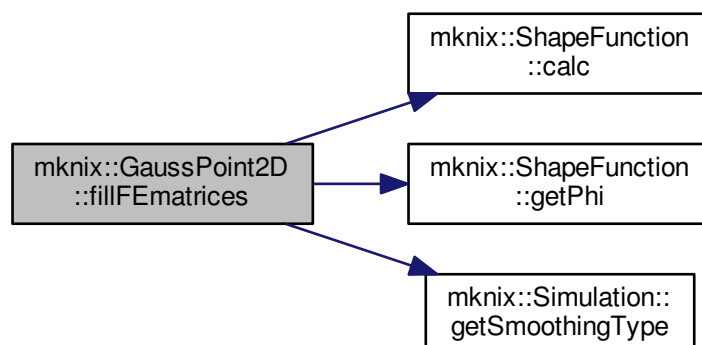


6.33.3.17 `void mknix::GaussPoint2D::fillFEmatrices ( )` [override],[virtual]

Implements [mknix::GaussPoint](#).

Definition at line 77 of file `gausspoint2D.cpp`.

Here is the call graph for this function:

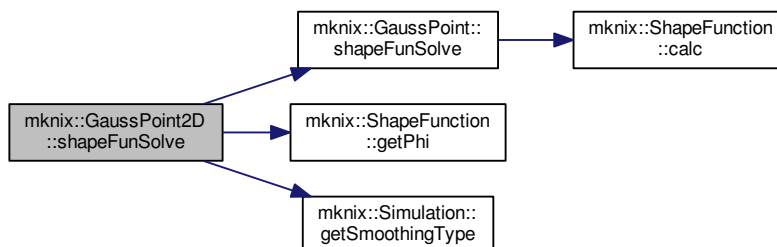


**6.33.3.18** `void mknix::GaussPoint2D::shapeFunSolve ( std::string type_in, double q_in )` [override],[virtual]

Reimplemented from [mknix::GaussPoint](#).

Definition at line 45 of file `gausspoint2D.cpp`.

Here is the call graph for this function:



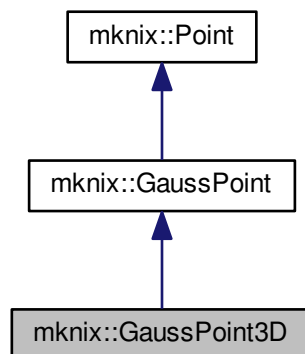
The documentation for this class was generated from the following files:

- [gausspoint2D.h](#)
- [gausspoint2D.cpp](#)

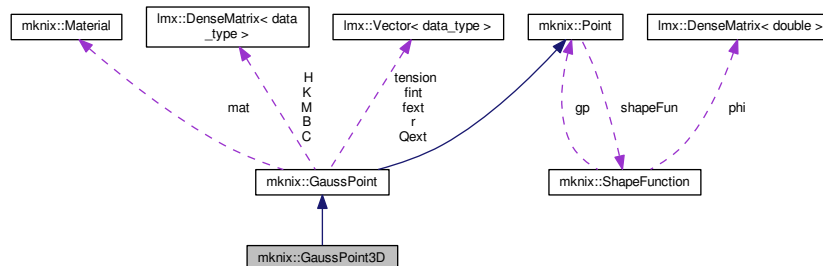
## 6.34 mknix::GaussPoint3D Class Reference

```
#include <gausspoint3D.h>
```

Inheritance diagram for mknix::GaussPoint3D:



Collaboration diagram for mknix::GaussPoint3D:



## Public Member Functions

- [GaussPoint3D](#) ()
- [GaussPoint3D](#) (double alpha\_in, double weight\_in, double jacobian\_in, [Material](#) \*mat\_in, int num, double coor\_x, double coor\_y, double coor\_z, double dc\_in, bool stressPoint\_in)
- [~GaussPoint3D](#) ()
- void [shapeFunSolve](#) (std::string, double) override
- void [fillFEmatrices](#) () override
- void [computeMij](#) () override
- void [computeKij](#) () override
- void [computeStress](#) () override
- void [computeNLStress](#) () override
- void [computeFint](#) () override
- void [computeFext](#) () override
- void [computeNLFint](#) () override
- void [computeNLKij](#) () override
- void [assembleMij](#) (Imx::Matrix< data\_type > &) override
- void [assembleKij](#) (Imx::Matrix< data\_type > &) override
- void [assembleRi](#) (Imx::Vector< data\_type > &, int) override

- void [assembleFint](#) (Imx::Vector< data\_type > &) override
- void [assembleFext](#) (Imx::Vector< data\_type > &) override
- double [calcPotentialE](#) (const Imx::Vector< data\_type > &) override
- double [calcKineticE](#) (const Imx::Vector< data\_type > &) override
- double [calcElasticE](#) () override

#### Additional Inherited Members

##### 6.34.1 Detailed Description

###### Author

Daniel Iglesias

Definition at line 45 of file gausspoint3D.h.

##### 6.34.2 Constructor & Destructor Documentation

###### 6.34.2.1 mknix::GaussPoint3D::GaussPoint3D ( )

Definition at line 13 of file gausspoint3D.cpp.

6.34.2.2 mknix::GaussPoint3D::GaussPoint3D ( double *alpha\_in*, double *weight\_in*, double *jacobian\_in*, Material \* *mat\_in*, int *num*, double *coor\_x*, double *coor\_y*, double *coor\_z*, double *dc\_in*, bool *stressPoint\_in* )

Definition at line 18 of file gausspoint3D.cpp.

###### 6.34.2.3 mknix::GaussPoint3D::~~GaussPoint3D ( )

Definition at line 46 of file gausspoint3D.cpp.

##### 6.34.3 Member Function Documentation

6.34.3.1 void mknix::GaussPoint3D::assembleFext ( Imx::Vector< data\_type > & *globalFext* ) [override],  
[virtual]

Implements [mknix::GaussPoint](#).

Definition at line 566 of file gausspoint3D.cpp.

6.34.3.2 void mknix::GaussPoint3D::assembleFint ( Imx::Vector< data\_type > & *globalFint* ) [override],  
[virtual]

Implements [mknix::GaussPoint](#).

Definition at line 554 of file gausspoint3D.cpp.

6.34.3.3 void mknix::GaussPoint3D::assembleKij ( Imx::Matrix< data\_type > & *globalTangent* ) [override],  
[virtual]

Implements [mknix::GaussPoint](#).

Definition at line 519 of file gausspoint3D.cpp.

6.34.3.4 void mknix::GaussPoint3D::assembleMij ( Imx::Matrix< data\_type > & *globalMass* ) [override],  
[virtual]

Implements [mknix::GaussPoint](#).

Definition at line 497 of file gausspoint3D.cpp.



6.34.3.5 `void mknix::GaussPoint3D::assembleRi ( Imx::Vector< data_type > & bodyR, int firstNode ) [override], [virtual]`

Implements [mknix::GaussPoint](#).

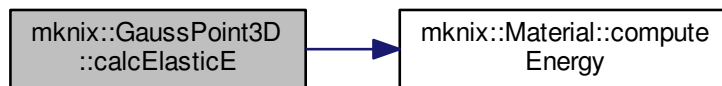
Definition at line 539 of file gausspoint3D.cpp.

6.34.3.6 `double mknix::GaussPoint3D::calcElasticE ( ) [override], [virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 609 of file gausspoint3D.cpp.

Here is the call graph for this function:



6.34.3.7 `double mknix::GaussPoint3D::calcKineticE ( const Imx::Vector< data_type > & qdot ) [override], [virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 591 of file gausspoint3D.cpp.

Here is the call graph for this function:

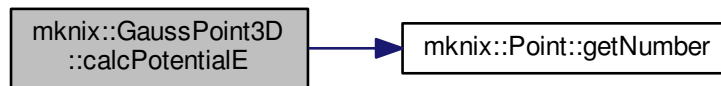


6.34.3.8 `double mknix::GaussPoint3D::calcPotentialE ( const Imx::Vector< data_type > & q ) [override], [virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 578 of file gausspoint3D.cpp.

Here is the call graph for this function:

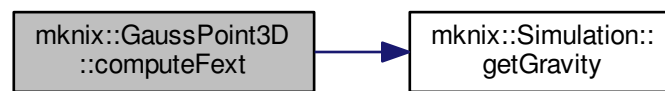


**6.34.3.9** `void mknix::GaussPoint3D::computeFext ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 285 of file `gausspoint3D.cpp`.

Here is the call graph for this function:

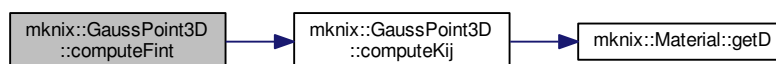


**6.34.3.10** `void mknix::GaussPoint3D::computeFint ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 269 of file `gausspoint3D.cpp`.

Here is the call graph for this function:

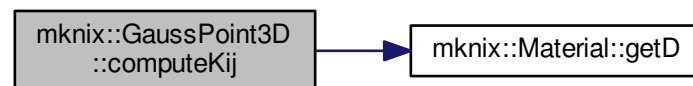


**6.34.3.11** `void mknix::GaussPoint3D::computeKij ( ) [override],[virtual]`

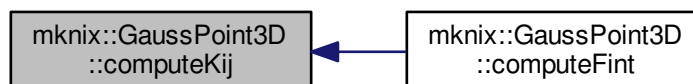
Implements [mknix::GaussPoint](#).

Definition at line 137 of file `gausspoint3D.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:

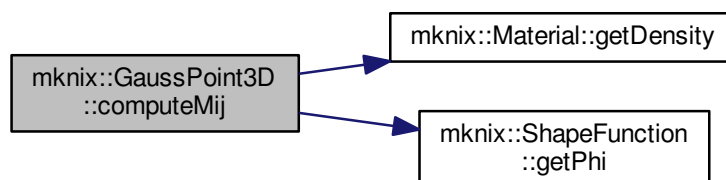


**6.34.3.12** `void mknix::GaussPoint3D::computeMij ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 114 of file `gausspoint3D.cpp`.

Here is the call graph for this function:

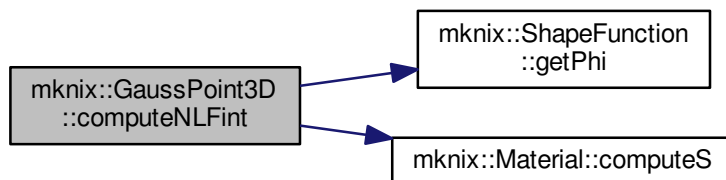


**6.34.3.13** `void mknix::GaussPoint3D::computeNLFint ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 302 of file `gausspoint3D.cpp`.

Here is the call graph for this function:

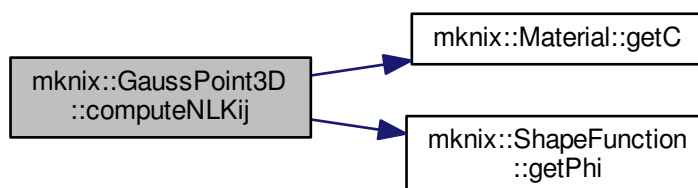


**6.34.3.14** `void mknix::GaussPoint3D::computeNLKij ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 326 of file `gausspoint3D.cpp`.

Here is the call graph for this function:

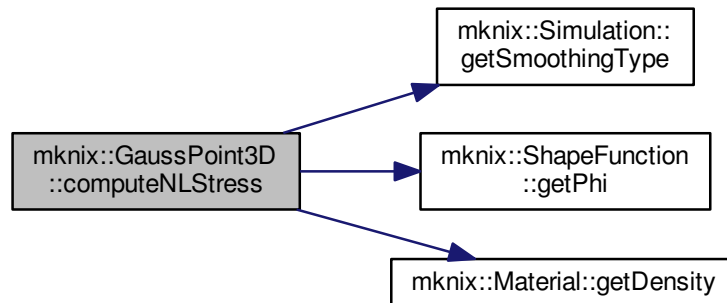


**6.34.3.15** `void mknix::GaussPoint3D::computeNLStress ( ) [override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 212 of file `gausspoint3D.cpp`.

Here is the call graph for this function:

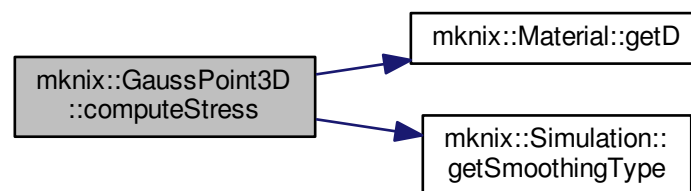


**6.34.3.16** `void mknix::GaussPoint3D::computeStress ( )` `[override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 167 of file `gausspoint3D.cpp`.

Here is the call graph for this function:

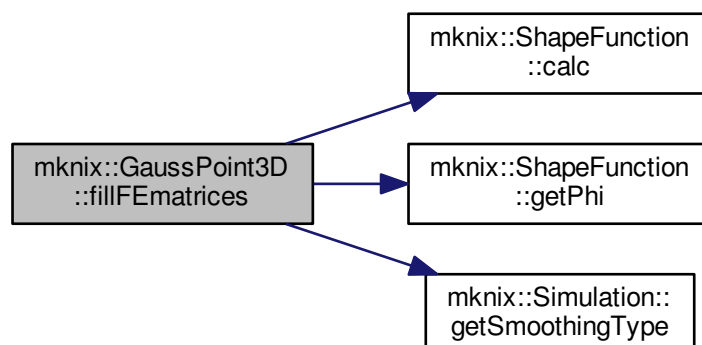


**6.34.3.17** `void mknix::GaussPoint3D::fillFEmatrices ( )` `[override],[virtual]`

Implements [mknix::GaussPoint](#).

Definition at line 82 of file `gausspoint3D.cpp`.

Here is the call graph for this function:

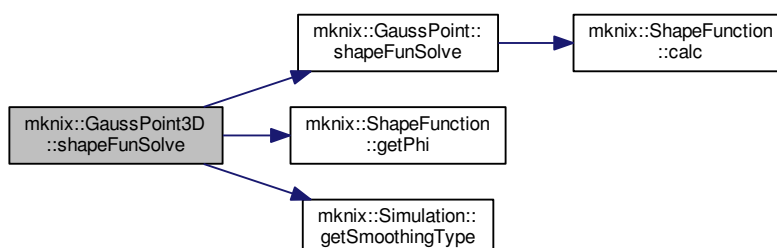


**6.34.3.18** `void mknix::GaussPoint3D::shapeFunSolve ( std::string type_in, double q_in )` [override],[virtual]

Reimplemented from [mknix::GaussPoint](#).

Definition at line 51 of file `gausspoint3D.cpp`.

Here is the call graph for this function:



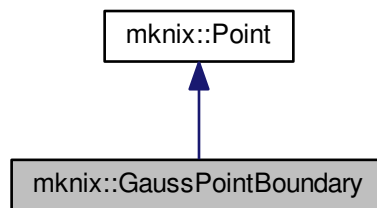
The documentation for this class was generated from the following files:

- [gausspoint3D.h](#)
- [gausspoint3D.cpp](#)

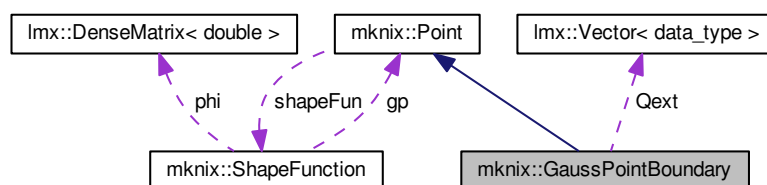
## 6.35 mknix::GaussPointBoundary Class Reference

```
#include <gausspointboundary.h>
```

Inheritance diagram for mknix::GaussPointBoundary:



Collaboration diagram for mknix::GaussPointBoundary:



#### Public Member Functions

- [GaussPointBoundary](#) ()
- [GaussPointBoundary](#) (int dim\_in, double alpha\_in, double weight\_in, double jacobian\_in, int num\_in, double coor\_x, double dc\_in)
- [GaussPointBoundary](#) (int dim\_in, double alpha\_in, double weight\_in, double jacobian\_in, int num\_in, double coor\_x, double coor\_y, double dc\_in)
- [~GaussPointBoundary](#) ()
- virtual void [shapeFunSolve](#) (std::string, double) override
- void [computeQext](#) ([LoadThermalBoundary1D](#) \*)
- void [assembleQext](#) ([Imx::Vector< data\\_type >](#) &)

#### Protected Member Functions

- void [initializeMatVecs](#) ()

#### Protected Attributes

- int [num](#)
- double [weight](#)
- [Imx::Vector< data\\_type >](#) [Qext](#)

### 6.35.1 Detailed Description

#### Author

Daniel Iglesias

Definition at line 44 of file gausspointboundary.h.

### 6.35.2 Constructor & Destructor Documentation

#### 6.35.2.1 `mknix::GaussPointBoundary::GaussPointBoundary ( )`

Definition at line 14 of file gausspointboundary.cpp.

#### 6.35.2.2 `mknix::GaussPointBoundary::GaussPointBoundary ( int dim_in, double alpha_in, double weight_in, double jacobian_in, int num_in, double coor_x, double dc_in )`

Definition at line 19 of file gausspointboundary.cpp.

#### 6.35.2.3 `mknix::GaussPointBoundary::GaussPointBoundary ( int dim_in, double alpha_in, double weight_in, double jacobian_in, int num_in, double coor_x, double coor_y, double dc_in )`

Definition at line 32 of file gausspointboundary.cpp.

#### 6.35.2.4 `mknix::GaussPointBoundary::~~GaussPointBoundary ( )`

Definition at line 46 of file gausspointboundary.cpp.

### 6.35.3 Member Function Documentation

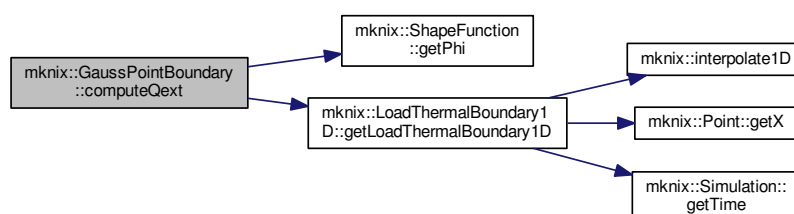
#### 6.35.3.1 `void mknix::GaussPointBoundary::assembleQext ( Imx::Vector< data_type > & globalHeat )`

Definition at line 103 of file gausspointboundary.cpp.

#### 6.35.3.2 `void mknix::GaussPointBoundary::computeQext ( LoadThermalBoundary1D * loadThermalBoundary_in )`

Definition at line 88 of file gausspointboundary.cpp.

Here is the call graph for this function:

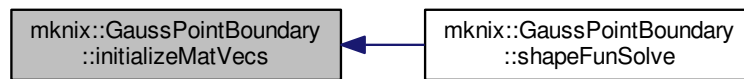


#### 6.35.3.3 `void mknix::GaussPointBoundary::initializeMatVecs ( )` [protected]

Definition at line 119 of file gausspointboundary.cpp.



Here is the caller graph for this function:

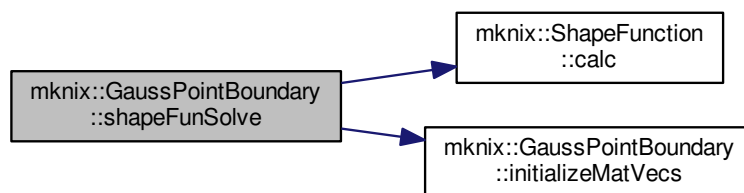


6.35.3.4 `void mknix::GaussPointBoundary::shapeFunSolve ( std::string type_in, double q_in ) [override], [virtual]`

Reimplemented from [mknix::Point](#).

Definition at line 51 of file `gausspointboundary.cpp`.

Here is the call graph for this function:



#### 6.35.4 Member Data Documentation

6.35.4.1 `int mknix::GaussPointBoundary::num [protected]`

Definition at line 79 of file `gausspointboundary.h`.

6.35.4.2 `Imx::Vector<data_type> mknix::GaussPointBoundary::Qext [protected]`

Definition at line 82 of file `gausspointboundary.h`.

6.35.4.3 `double mknix::GaussPointBoundary::weight [protected]`

Definition at line 80 of file `gausspointboundary.h`.

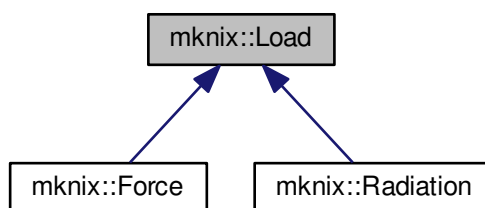
The documentation for this class was generated from the following files:

- [gausspointboundary.h](#)
- [gausspointboundary.cpp](#)

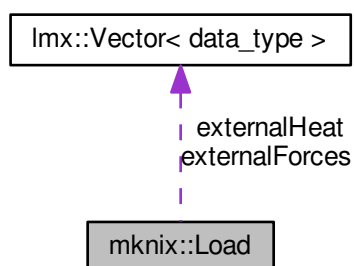
#### 6.36 mknix::Load Class Reference

```
#include <load.h>
```

Inheritance diagram for mknix::Load:



Collaboration diagram for mknix::Load:



#### Public Member Functions

- `Load ()`
- virtual `~Load ()`
- virtual void `assembleExternalForces (Imx::Vector< data_type > &)`
- virtual void `outputToFile (std::ofstream *)=0`

#### Protected Attributes

- `std::vector< Node * > nodes`
- `Imx::Vector< data_type > externalForces`
- `Imx::Vector< data_type > externalHeat`

#### 6.36.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 32 of file `load.h`.

### 6.36.2 Constructor & Destructor Documentation

#### 6.36.2.1 mknix::Load::Load ( )

Definition at line 27 of file load.cpp.

#### 6.36.2.2 mknix::Load::~~Load ( ) [virtual]

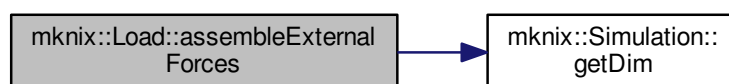
Definition at line 32 of file load.cpp.

### 6.36.3 Member Function Documentation

#### 6.36.3.1 void mknix::Load::assembleExternalForces ( Imx::Vector< data\_type > & globalExternalForces ) [virtual]

Definition at line 38 of file load.cpp.

Here is the call graph for this function:



#### 6.36.3.2 virtual void mknix::Load::outputToFile ( std::ofstream \* ) [pure virtual]

Implemented in [mknix::Radiation](#), and [mknix::Force](#).

### 6.36.4 Member Data Documentation

#### 6.36.4.1 Imx::Vector<data\_type> mknix::Load::externalForces [protected]

Definition at line 44 of file load.h.

#### 6.36.4.2 Imx::Vector<data\_type> mknix::Load::externalHeat [protected]

Definition at line 45 of file load.h.

#### 6.36.4.3 std::vector<Node\*> mknix::Load::nodes [protected]

Definition at line 43 of file load.h.

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

- [load.h](#)
- [load.cpp](#)

## 6.37 mknix::LoadThermal Class Reference

```
#include <loadthermal.h>
```

## Public Member Functions

- [LoadThermal](#) ()
- [LoadThermal](#) ([Node](#) \*, double)
- virtual [~LoadThermal](#) ()
- virtual void [insertNodesXCoordinates](#) (std::vector< double > &)
- virtual void [updateLoad](#) (double load\_in)
- virtual void [assembleExternalHeat](#) (Imx::Vector< [data\\_type](#) > &)
- virtual void [outputToFile](#) (std::ofstream \*)
- void [getMaxTemp](#) (double &)

## Protected Attributes

- std::vector< [Node](#) \* > [nodes](#)
- [data\\_type](#) [externalHeat](#)

### 6.37.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 32 of file loadthermal.h.

### 6.37.2 Constructor & Destructor Documentation

#### 6.37.2.1 `mknix::LoadThermal::LoadThermal ( )`

Definition at line 27 of file loadthermal.cpp.

#### 6.37.2.2 `mknix::LoadThermal::LoadThermal ( Node * node_in, double fluence_in )`

Definition at line 31 of file loadthermal.cpp.

#### 6.37.2.3 `mknix::LoadThermal::~~LoadThermal ( )` `[virtual]`

Definition at line 37 of file loadthermal.cpp.

### 6.37.3 Member Function Documentation

#### 6.37.3.1 `void mknix::LoadThermal::assembleExternalHeat ( Imx::Vector< data_type > & globalExternalHeat )` `[virtual]`

Definition at line 53 of file loadthermal.cpp.

#### 6.37.3.2 `void mknix::LoadThermal::getMaxTemp ( double & maxTemp_in )`

Definition at line 66 of file loadthermal.cpp.

#### 6.37.3.3 `void mknix::LoadThermal::insertNodesXCoordinates ( std::vector< double > & x_coordinates )` `[virtual]`

Definition at line 41 of file loadthermal.cpp.

#### 6.37.3.4 `virtual void mknix::LoadThermal::outputToFile ( std::ofstream * )` `[inline]`, `[virtual]`

Definition at line 47 of file loadthermal.h.

6.37.3.5 `virtual void mknix::LoadThermal::updateLoad ( double load_in ) [inline],[virtual]`

Definition at line 42 of file `loadthermal.h`.

#### 6.37.4 Member Data Documentation

6.37.4.1 `data_type mknix::LoadThermal::externalHeat [protected]`

Definition at line 54 of file `loadthermal.h`.

6.37.4.2 `std::vector<Node*> mknix::LoadThermal::nodes [protected]`

Definition at line 53 of file `loadthermal.h`.

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

- [loadthermal.h](#)
- [loadthermal.cpp](#)

## 6.38 mknix::LoadThermalBody Class Reference

```
#include <loadthermalbody.h>
```

#### Public Member Functions

- [LoadThermalBody \(\)](#)
- [~LoadThermalBody \(\)](#)
- `double` [getLoadThermalBody \(Point \\*\)](#)

#### Protected Attributes

- `std::map< double, double >` [srim](#)

#### 6.38.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 34 of file `loadthermalbody.h`.

#### 6.38.2 Constructor & Destructor Documentation

6.38.2.1 `mknix::LoadThermalBody::LoadThermalBody ( )`

Definition at line 28 of file `loadthermalbody.cpp`.

6.38.2.2 `mknix::LoadThermalBody::~~LoadThermalBody ( )`

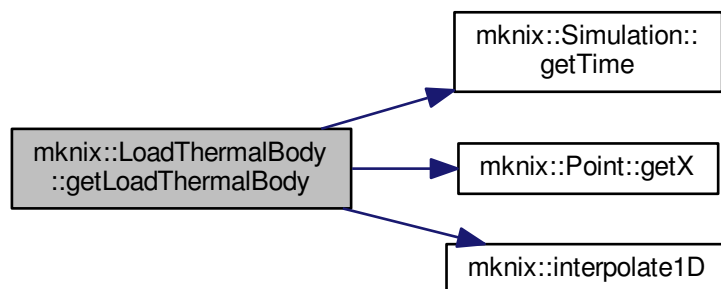
Definition at line 48 of file `loadthermalbody.cpp`.

### 6.38.3 Member Function Documentation

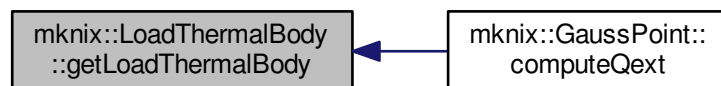
#### 6.38.3.1 `double mknix::LoadThermalBody::getLoadThermalBody ( Point * thePoint )`

Definition at line 52 of file loadthermalbody.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.38.4 Member Data Documentation

#### 6.38.4.1 `std::map<double, double> mknix::LoadThermalBody::srin` [protected]

Definition at line 45 of file loadthermalbody.h.

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

- [loadthermalbody.h](#)
- [loadthermalbody.cpp](#)

## 6.39 mknix::LoadThermalBoundary1D Class Reference

```
#include <loadthermalboundary1D.h>
```

### Public Member Functions

- [LoadThermalBoundary1D](#) ()

- [~LoadThermalBoundary1D](#) ()
- void [loadFile](#) (std::string)
- void [loadTimeFile](#) (std::string)
- void [scaleLoad](#) (double)
- double [getLoadThermalBoundary1D](#) (Point \*)

#### Protected Attributes

- std::map< double, double > [loadmap](#)
- std::map< double, double > [timemap](#)

#### 6.39.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 34 of file loadthermalboundary1D.h.

#### 6.39.2 Constructor & Destructor Documentation

##### 6.39.2.1 mknix::LoadThermalBoundary1D::LoadThermalBoundary1D ( )

Definition at line 28 of file loadthermalboundary1D.cpp.

##### 6.39.2.2 mknix::LoadThermalBoundary1D::~~LoadThermalBoundary1D ( )

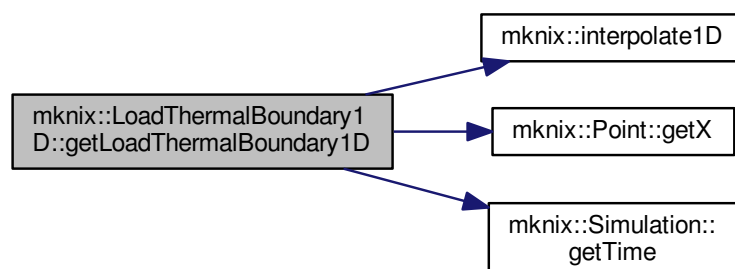
Definition at line 33 of file loadthermalboundary1D.cpp.

#### 6.39.3 Member Function Documentation

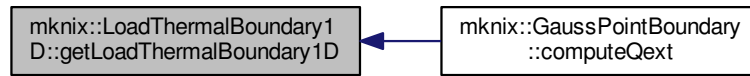
##### 6.39.3.1 double mknix::LoadThermalBoundary1D::getLoadThermalBoundary1D ( Point \* *thePoint* )

Definition at line 78 of file loadthermalboundary1D.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.39.3.2 void mknix::LoadThermalBoundary1D::loadFile ( std::string *fileName* )

Definition at line 37 of file loadthermalboundary1D.cpp.

#### 6.39.3.3 void mknix::LoadThermalBoundary1D::loadTimeFile ( std::string *fileName* )

Definition at line 53 of file loadthermalboundary1D.cpp.

#### 6.39.3.4 void mknix::LoadThermalBoundary1D::scaleLoad ( double *loadFactor\_in* )

Definition at line 69 of file loadthermalboundary1D.cpp.

### 6.39.4 Member Data Documentation

#### 6.39.4.1 std::map<double, double> mknix::LoadThermalBoundary1D::loadmap [protected]

Definition at line 51 of file loadthermalboundary1D.h.

#### 6.39.4.2 std::map<double, double> mknix::LoadThermalBoundary1D::timemap [protected]

Definition at line 52 of file loadthermalboundary1D.h.

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

- [loadthermalboundary1D.h](#)
- [loadthermalboundary1D.cpp](#)

## 6.40 mknix::Material Class Reference

```
#include <material.h>
```

### Public Member Functions

- [Material](#) ()
- [~Material](#) ()
- double [getE](#) ()
- double [getMu](#) ()
- double [getDensity](#) ()
- double [getCapacity](#) (double temp\_in=0)
- double [getKappa](#) (double temp\_in=0)
- [Imx::DenseMatrix](#)< double > & [getD](#) ()
- [Imx::DenseMatrix](#)< double > & [getC](#) ()
- void [setThermalProps](#) (double capacity\_in, double kappa\_in, double beta\_in, double density\_in)
- void [setMechanicalProps](#) (int dim\_in, double young\_in, double poisson\_in, double density\_in)



- void [addThermalCapacity](#) (double temp\_in, double capacity\_in)
- void [addThermalConductivity](#) (double temp\_in, double conductivity\_in)
- void [computeD](#) ()
- void [computeC](#) ()
- double [getCsym](#) (int &i, int &j, int &k, int &l)
- void [computeS](#) (cofe::TensorRank2Sym< 2, double > &S, const cofe::TensorRank2< 2, double > &F, double)
- void [computeS](#) (cofe::TensorRank2Sym< 3, double > &S, const cofe::TensorRank2< 3, double > &F)
- double [computeEnergy](#) (const cofe::TensorRank2< 2, double > &S)
- double [computeEnergy](#) (const cofe::TensorRank2< 3, double > &S)

#### 6.40.1 Detailed Description

##### Author

Daniel Iglesias

Definition at line 32 of file material.h.

#### 6.40.2 Constructor & Destructor Documentation

##### 6.40.2.1 mknix::Material::Material ( )

Definition at line 24 of file material.cpp.

##### 6.40.2.2 mknix::Material::~~Material ( )

Definition at line 35 of file material.cpp.

#### 6.40.3 Member Function Documentation

##### 6.40.3.1 void mknix::Material::addThermalCapacity ( double *temp\_in*, double *capacity\_in* ) [inline]

Definition at line 81 of file material.h.

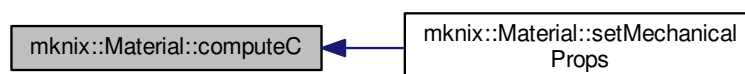
##### 6.40.3.2 void mknix::Material::addThermalConductivity ( double *temp\_in*, double *conductivity\_in* ) [inline]

Definition at line 83 of file material.h.

##### 6.40.3.3 void mknix::Material::computeC ( )

Definition at line 94 of file material.cpp.

Here is the caller graph for this function:



#### 6.40.3.4 void mknix::Material::computeD ( )

Definition at line 60 of file material.cpp.

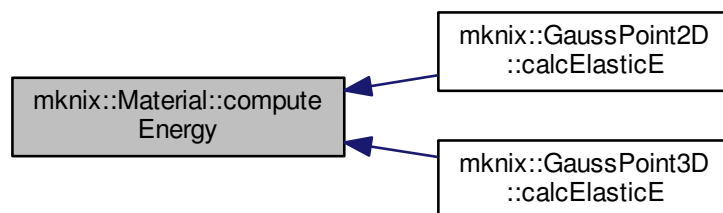
Here is the caller graph for this function:



#### 6.40.3.5 double mknix::Material::computeEnergy ( const cofe::TensorRank2< 2, double > & S )

Definition at line 183 of file material.cpp.

Here is the caller graph for this function:



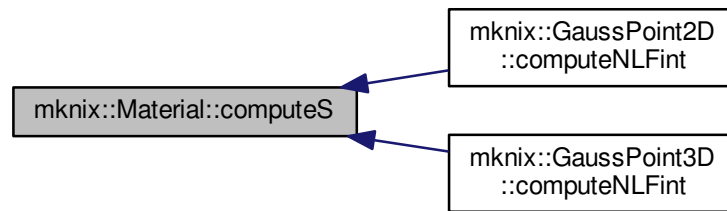
#### 6.40.3.6 double mknix::Material::computeEnergy ( const cofe::TensorRank2< 3, double > & S )

Definition at line 197 of file material.cpp.

#### 6.40.3.7 void mknix::Material::computeS ( cofe::TensorRank2Sym< 2, double > & S, const cofe::TensorRank2< 2, double > & F, double temperature\_in )

Definition at line 139 of file material.cpp.

Here is the caller graph for this function:



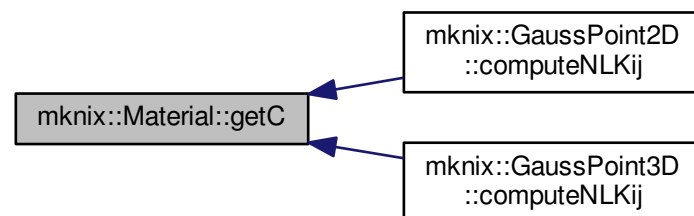
6.40.3.8 `void mknix::Material::computeS ( cofe::TensorRank2Sym< 3, double > & S, const cofe::TensorRank2< 3, double > & F )`

Definition at line 165 of file material.cpp.

6.40.3.9 `Imx::DenseMatrix<double>& mknix::Material::getC ( ) [inline]`

Definition at line 74 of file material.h.

Here is the caller graph for this function:



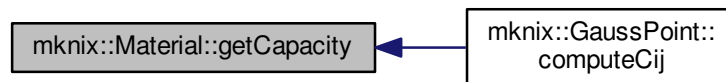
6.40.3.10 `double mknix::Material::getCapacity ( double temp_in = 0 ) [inline]`

Definition at line 63 of file material.h.

Here is the call graph for this function:



Here is the caller graph for this function:



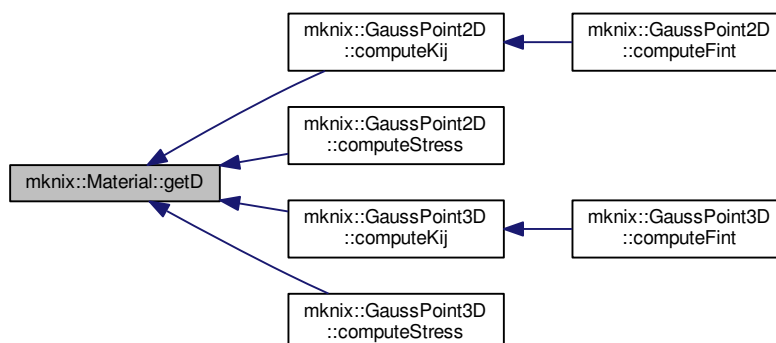
6.40.3.11 `double mknix::Material::getCsym ( int & i, int & j, int & k, int & l )`

Definition at line 126 of file material.cpp.

6.40.3.12 `Imx::DenseMatrix<double>& mknix::Material::getD ( ) [inline]`

Definition at line 71 of file material.h.

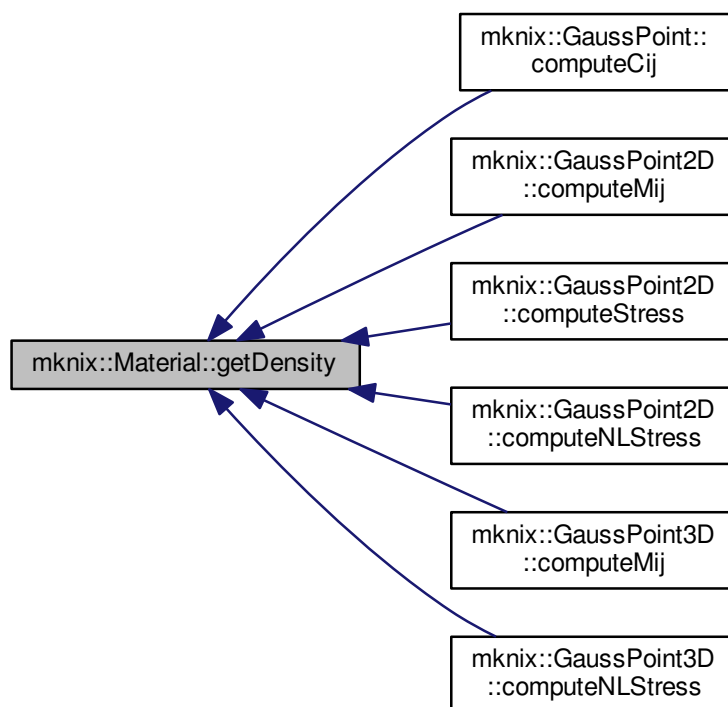
Here is the caller graph for this function:



6.40.3.13 `double mknix::Material::getDensity ( ) [inline]`

Definition at line 60 of file material.h.

Here is the caller graph for this function:



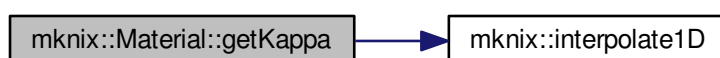
6.40.3.14 `double mknix::Material::getE ( ) [inline]`

Definition at line 54 of file `material.h`.

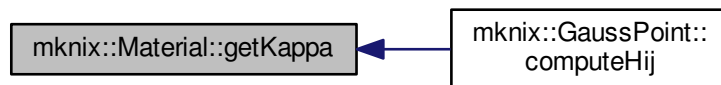
6.40.3.15 `double mknix::Material::getKappa ( double temp_in = 0 ) [inline]`

Definition at line 67 of file `material.h`.

Here is the call graph for this function:



Here is the caller graph for this function:



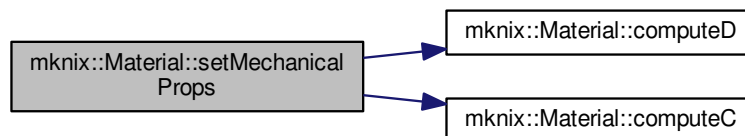
6.40.3.16 `double mknix::Material::getMu ( ) [inline]`

Definition at line 57 of file `material.h`.

6.40.3.17 `void mknix::Material::setMechanicalProps ( int dim_in, double young_in, double poisson_in, double density_in )`

Definition at line 48 of file `material.cpp`.

Here is the call graph for this function:



6.40.3.18 `void mknix::Material::setThermalProps ( double capacity_in, double kappa_in, double beta_in, double density_in )`

Definition at line 39 of file `material.cpp`.

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

- [material.h](#)
- [material.cpp](#)

## 6.41 `Imx::Matrix< T >` Class Template Reference

```
#include <cell.h>
```

### 6.41.1 Detailed Description

```
template<typename T>class Imx::Matrix< T >
```

Definition at line 40 of file `cell.h`.

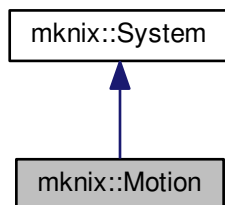
The documentation for this class was generated from the following file:

- [cell.h](#)

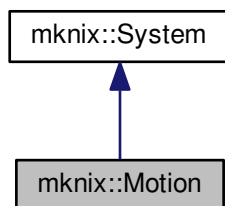
## 6.42 mknix::Motion Class Reference

```
#include <motion.h>
```

Inheritance diagram for mknix::Motion:



Collaboration diagram for mknix::Motion:



## Public Member Functions

- [Motion](#) ()
- [Motion](#) (Node \*)
- [~Motion](#) ()
- void [setNode](#) (Node \*node\_in)
- void [setTimeUx](#) (std::map< double, double > &timeUX\_in)
- void [setTimeUy](#) (std::map< double, double > &timeUY\_in)
- void [setTimeUz](#) (std::map< double, double > &timeUZ\_in)
- void [update](#) (double)

## Additional Inherited Members

## 6.42.1 Detailed Description

## Author

AUTHORS <MAILS>

Definition at line 32 of file `motion.h`.

### 6.42.2 Constructor & Destructor Documentation

#### 6.42.2.1 `mknix::Motion::Motion ( )`

Definition at line 26 of file `motion.cpp`.

#### 6.42.2.2 `mknix::Motion::Motion ( Node * node_in )`

Definition at line 30 of file `motion.cpp`.

#### 6.42.2.3 `mknix::Motion::~~Motion ( )`

Definition at line 38 of file `motion.cpp`.

### 6.42.3 Member Function Documentation

#### 6.42.3.1 `void mknix::Motion::setNode ( Node * node_in ) [inline]`

Definition at line 42 of file `motion.h`.

#### 6.42.3.2 `void mknix::Motion::setTimeUx ( std::map< double, double > & timeUX_in ) [inline]`

Definition at line 48 of file `motion.h`.

#### 6.42.3.3 `void mknix::Motion::setTimeUy ( std::map< double, double > & timeUY_in ) [inline]`

Definition at line 53 of file `motion.h`.

#### 6.42.3.4 `void mknix::Motion::setTimeUz ( std::map< double, double > & timeUZ_in ) [inline]`

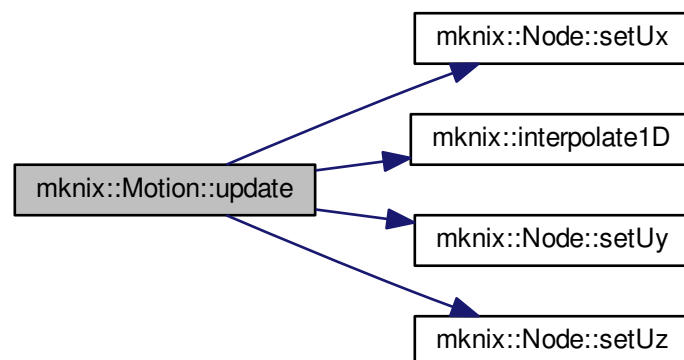
Definition at line 58 of file `motion.h`.

#### 6.42.3.5 `void mknix::Motion::update ( double theTime ) [virtual]`

Reimplemented from [mknix::System](#).

Definition at line 42 of file `motion.cpp`.

Here is the call graph for this function:





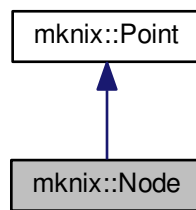
The documentation for this class was generated from the following files:

- [motion.h](#)
- [motion.cpp](#)

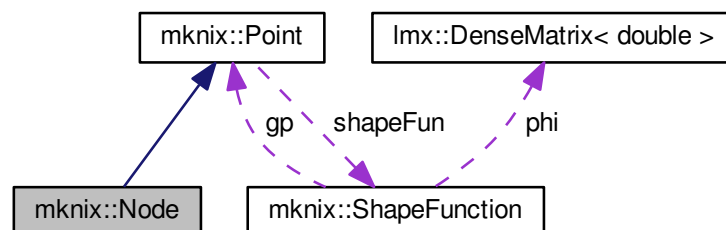
## 6.43 mknix::Node Class Reference

```
#include <node.h>
```

Inheritance diagram for mknix::Node:



Collaboration diagram for mknix::Node:



### Public Member Functions

- [Node](#) ()
- [Node](#) (const [Node](#) &)
- [Node](#) (const [Node](#) \*)
- [Node](#) (int i\_in, double x\_in, double y\_in, double z\_in)
- [~Node](#) ()
- void [addWeight](#) (double w\_in)
- const double & [getWeight](#) () const
- const int & [getThermalNumber](#) () const
- const void [setNumber](#) (int num\_in)
- const void [setThermalNumber](#) (int num\_in)

- const double & [getqt](#) () const
- double [getqx](#) (int i) const
- double [getUx](#) () const
- double [getUy](#) () const
- double [getUz](#) () const
- double [getU](#) (int i) const
- double [getConf](#) (int dof) const override
- double [getTemp](#) () const override
- size\_t [getSupportSize](#) (int deriv)
- int [getSupportNodeNumber](#) (int deriv, int s\_node)
- double [getShapeFunValue](#) (int deriv, int s\_node)
- void [setUx](#) (double ux\_in)
- void [setUy](#) (double uy\_in)
- void [setUz](#) (double uz\_in)
- void [setqx](#) (const [lmx::Vector](#)< [data\\_type](#) > &globalConf, int [dim](#))
- void [setqt](#) (const [lmx::Vector](#)< [data\\_type](#) > &globalConf)
- void [setqt](#) (double &temp\_in)
- void [setX](#) (double X\_in)
- void [setY](#) (double Y\_in)
- void [setZ](#) (double Z\_in)

#### Additional Inherited Members

##### 6.43.1 Detailed Description

###### Author

AUTHORS <MAILS>

Definition at line 31 of file node.h.

##### 6.43.2 Constructor & Destructor Documentation

###### 6.43.2.1 `mknix::Node::Node ( )`

Definition at line 25 of file node.cpp.

###### 6.43.2.2 `mknix::Node::Node ( const Node & p_node_in )`

Definition at line 27 of file node.cpp.

###### 6.43.2.3 `mknix::Node::Node ( const Node * p_node_in )`

Definition at line 35 of file node.cpp.

###### 6.43.2.4 `mknix::Node::Node ( int i_in, double x_in, double y_in, double z_in )`

Definition at line 43 of file node.cpp.

###### 6.43.2.5 `mknix::Node::~~Node ( )`

Definition at line 53 of file node.cpp.

##### 6.43.3 Member Function Documentation

###### 6.43.3.1 `void mknix::Node::addWeight ( double w_in ) [inline]`

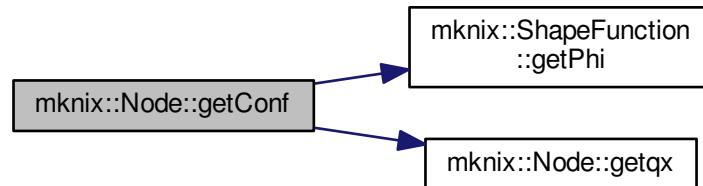
Definition at line 43 of file node.h.

6.43.3.2 `double mknix::Node::getConf ( int dof ) const` `[override],[virtual]`

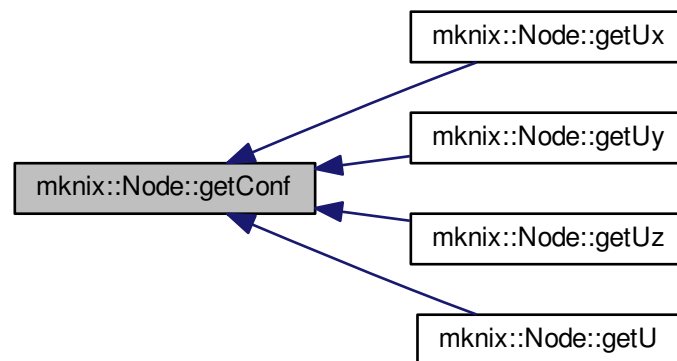
Reimplemented from [mknix::Point](#).

Definition at line 57 of file node.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.43.3.3 `const double& mknix::Node::getqt ( ) const` `[inline]`

Definition at line 75 of file node.h.

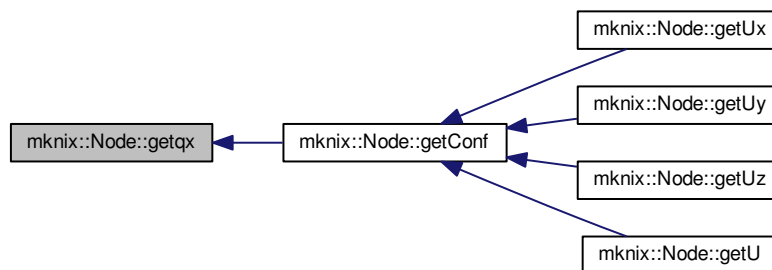
Here is the caller graph for this function:



#### 6.43.3.4 `double mknix::Node::getqx ( int i ) const [inline]`

Definition at line 79 of file node.h.

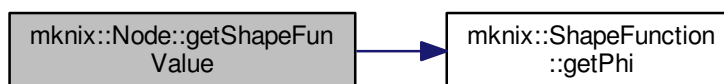
Here is the caller graph for this function:



#### 6.43.3.5 `double mknix::Node::getShapeFunValue ( int deriv, int s_node )`

Definition at line 156 of file node.cpp.

Here is the call graph for this function:



#### 6.43.3.6 `int mknix::Node::getSupportNodeNumber ( int deriv, int s_node ) [virtual]`

Reimplemented from [mknix::Point](#).

Definition at line 127 of file node.cpp.

Here is the call graph for this function:

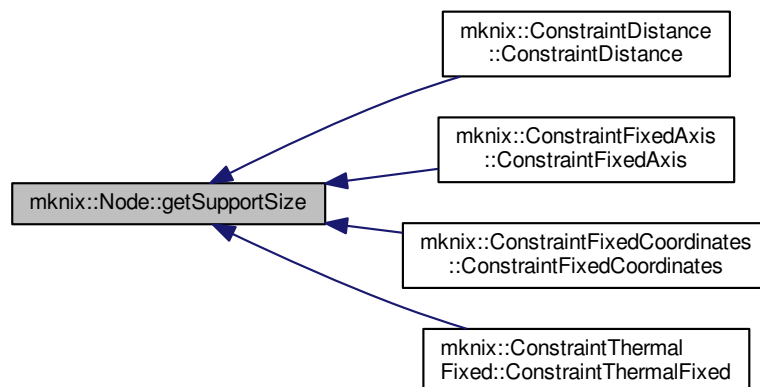


#### 6.43.3.7 `size_t mknix::Node::getSupportSize ( int deriv ) [virtual]`

Reimplemented from [mknix::Point](#).

Definition at line 100 of file node.cpp.

Here is the caller graph for this function:

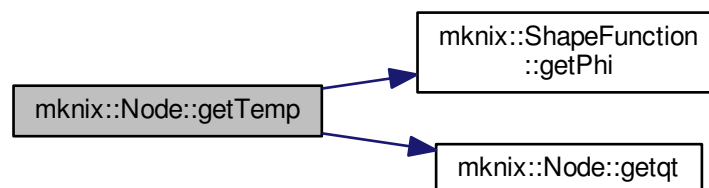


**6.43.3.8** `double mknix::Node::getTemp ( ) const` `[override], [virtual]`

Reimplemented from [mknix::Point](#).

Definition at line 80 of file node.cpp.

Here is the call graph for this function:



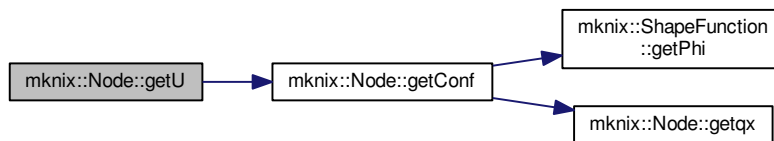
**6.43.3.9** `const int& mknix::Node::getThermalNumber ( ) const` `[inline]`

Definition at line 51 of file node.h.

**6.43.3.10** `double mknix::Node::getU ( int i ) const` `[inline]`

Definition at line 99 of file node.h.

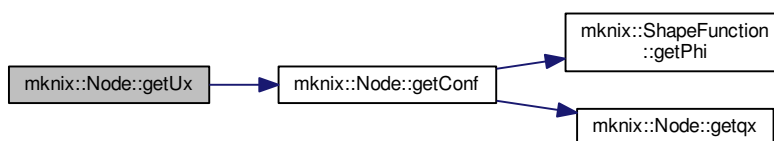
Here is the call graph for this function:



#### 6.43.3.11 `double mknix::Node::getUx ( ) const [inline]`

Definition at line 87 of file `node.h`.

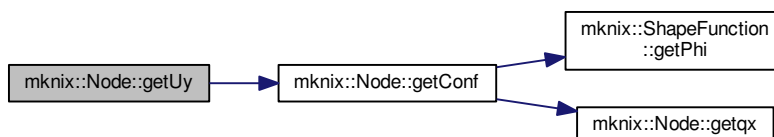
Here is the call graph for this function:



#### 6.43.3.12 `double mknix::Node::getUy ( ) const [inline]`

Definition at line 91 of file `node.h`.

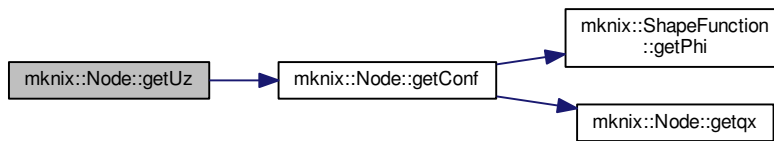
Here is the call graph for this function:



#### 6.43.3.13 `double mknix::Node::getUz ( ) const [inline]`

Definition at line 95 of file `node.h`.

Here is the call graph for this function:



6.43.3.14 `const double& mknix::Node::getWeight ( ) const` [inline]

Definition at line 47 of file node.h.

6.43.3.15 `const void mknix::Node::setNumber ( int num_in )` [inline]

Definition at line 55 of file node.h.

6.43.3.16 `void mknix::Node::setqt ( const Imx::Vector< data_type > & globalConf )`

Definition at line 177 of file node.cpp.

6.43.3.17 `void mknix::Node::setqt ( double & temp_in )` [inline]

Definition at line 134 of file node.h.

6.43.3.18 `void mknix::Node::setqx ( const Imx::Vector< data_type > & globalConf, int dim )`

Definition at line 168 of file node.cpp.

6.43.3.19 `const void mknix::Node::setThermalNumber ( int num_in )` [inline]

Definition at line 59 of file node.h.

6.43.3.20 `void mknix::Node::setUx ( double ux_in )` [inline]

Definition at line 118 of file node.h.

Here is the caller graph for this function:



6.43.3.21 `void mknix::Node::setUy ( double uy_in )` [inline]

Definition at line 122 of file node.h.

Here is the caller graph for this function:



**6.43.3.22** `void mknix::Node::setUz ( double uz_in ) [inline]`

Definition at line 126 of file node.h.

Here is the caller graph for this function:



**6.43.3.23** `void mknix::Node::setX ( double X_in ) [inline]`

Definition at line 149 of file node.h.

**6.43.3.24** `void mknix::Node::setY ( double Y_in ) [inline]`

Definition at line 154 of file node.h.

**6.43.3.25** `void mknix::Node::setZ ( double Z_in ) [inline]`

Definition at line 159 of file node.h.

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

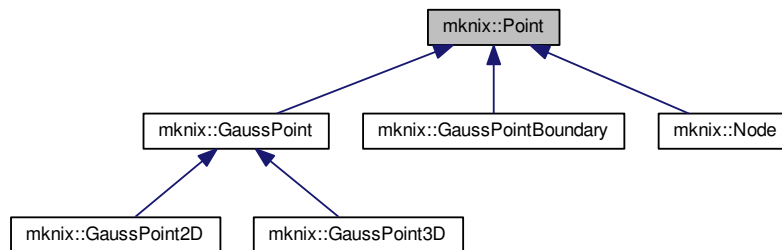
- [node.h](#)
- [node.cpp](#)

## 6.44 mknix::Point Class Reference

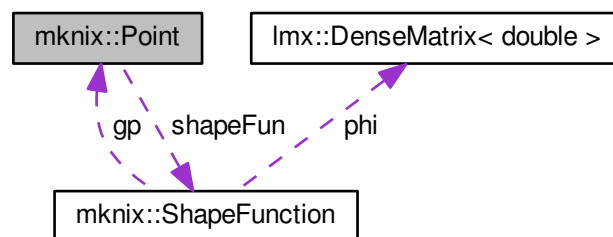
```
#include <point.h>
```



Inheritance diagram for mknix::Point:



Collaboration diagram for mknix::Point:



#### Public Member Functions

- [Point](#) ()
- [Point](#) (const [Point](#) &point\_in)
- [Point](#) (const [Point](#) \*point\_in)
- [Point](#) (int i, double coor\_x, double coor\_y, double coor\_z)
- [Point](#) (int dim, int i, double coor\_x, double coor\_y, double coor\_z, double alpha\_in, double dc\_in)
- [Point](#) (int dim, int i, double coor\_x, double coor\_y, double coor\_z, double jacobian\_in, double alpha\_in, double dc\_in)
- virtual [~Point](#) ()
- const int & [getDim](#) () const
- const double & [getX](#) () const
- const double & [getY](#) () const
- const double & [getZ](#) () const
- const int & [getNumber](#) () const
- virtual double [getTemp](#) () const
- virtual double [getConf](#) (int) const
- double [distance](#) ([Point](#) &) const
- void [setShapeFunType](#) (std::string type\_in)
- std::string [getShapeFunType](#) ()
- virtual size\_t [getSupportSize](#) (int deriv=0)
- virtual int [getSupportNodeNumber](#) (int deriv, int s\_node)

- void [setDc](#) (double dc\_in)
- void [setAlphai](#) (double alphai\_in)
- void [addSupportNode](#) ([Node](#) \*)
- void [findSupportNodes](#) (std::vector< [Node](#) \* > &)
- void [findSupportNodes](#) (std::vector< [Node](#) \* > &, double, double, double, double)
- virtual void [shapeFunSolve](#) (std::string, double)
- void [shapeFunSolve](#) (double q\_in)
- void [setJacobian](#) (double jac\_in)
- void [gnuplotOut](#) (std::ofstream &)
- const std::vector< [Node](#) \* > & [getSupportNodes](#) () const

#### Protected Attributes

- int [dim](#)
- int [num](#)
- double [X](#)
- double [Y](#)
- double [Z](#)
- double [alphai](#)
- double [dc](#)
- std::string [shapeFunType](#)
- [ShapeFunction](#) \* [shapeFun](#)
- std::vector< [Node](#) \* > [supportNodes](#)
- size\_t [supportNodesSize](#)
- double [jacobian](#)

#### Friends

- class [GaussPoint](#)
- class [ShapeFunction](#)
- class [ShapeFunctionRBF](#)
- class [ShapeFunctionMLS](#)
- class [ShapeFunctionTetrahedron](#)
- class [ShapeFunctionTriangle](#)
- class [ShapeFunctionLinear](#)
- class [ShapeFunctionLinearX](#)

#### 6.44.1 Detailed Description

##### Author

Daniel Iglesias

Definition at line 54 of file point.h.

#### 6.44.2 Constructor & Destructor Documentation

##### 6.44.2.1 [mnix::Point::Point](#) ( )

Definition at line 15 of file point.cpp.

##### 6.44.2.2 [mnix::Point::Point](#) ( const [Point](#) & *point\_in* )

Definition at line 19 of file point.cpp.

#### 6.44.2.3 mknix::Point::Point ( const Point \* point\_in )

Definition at line 35 of file point.cpp.

#### 6.44.2.4 mknix::Point::Point ( int i, double coor\_x, double coor\_y, double coor\_z )

Definition at line 51 of file point.cpp.

#### 6.44.2.5 mknix::Point::Point ( int dim, int i, double coor\_x, double coor\_y, double coor\_z, double alpha\_in, double dc\_in )

Definition at line 69 of file point.cpp.

#### 6.44.2.6 mknix::Point::Point ( int dim, int i, double coor\_x, double coor\_y, double coor\_z, double jacobian\_in, double alpha\_in, double dc\_in )

Definition at line 90 of file point.cpp.

#### 6.44.2.7 mknix::Point::~~Point ( ) [virtual]

Definition at line 112 of file point.cpp.

### 6.44.3 Member Function Documentation

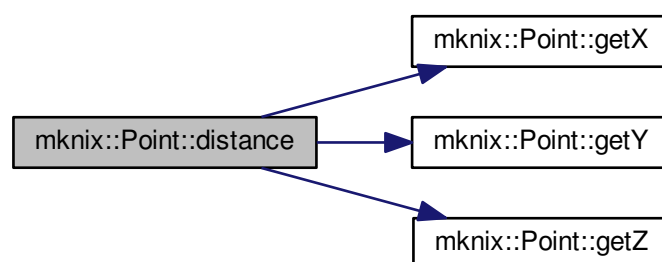
#### 6.44.3.1 void mknix::Point::addSupportNode ( Node \* node\_in )

Definition at line 148 of file point.cpp.

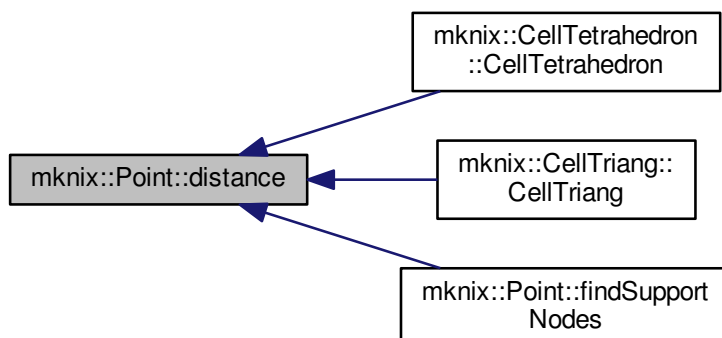
#### 6.44.3.2 double mknix::Point::distance ( Point & node\_in ) const

Definition at line 135 of file point.cpp.

Here is the call graph for this function:



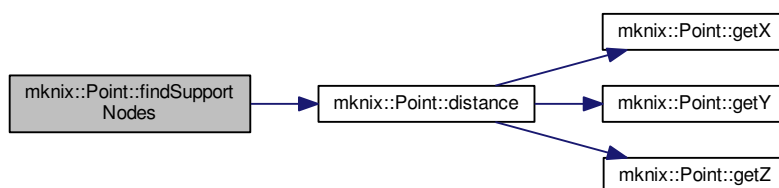
Here is the caller graph for this function:



#### 6.44.3.3 void mknix::Point::findSupportNodes ( std::vector< Node \* > & domainNodes )

Definition at line 155 of file point.cpp.

Here is the call graph for this function:



#### 6.44.3.4 void mknix::Point::findSupportNodes ( std::vector< Node \* > & domainNodes, double domain\_minX, double domain\_maxX, double domain\_minY, double domain\_maxY )

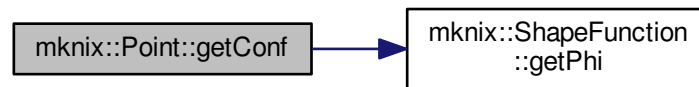
Definition at line 216 of file point.cpp.

#### 6.44.3.5 double mknix::Point::getConf ( int dof ) const [virtual]

Reimplemented in [mknix::Node](#).

Definition at line 126 of file point.cpp.

Here is the call graph for this function:



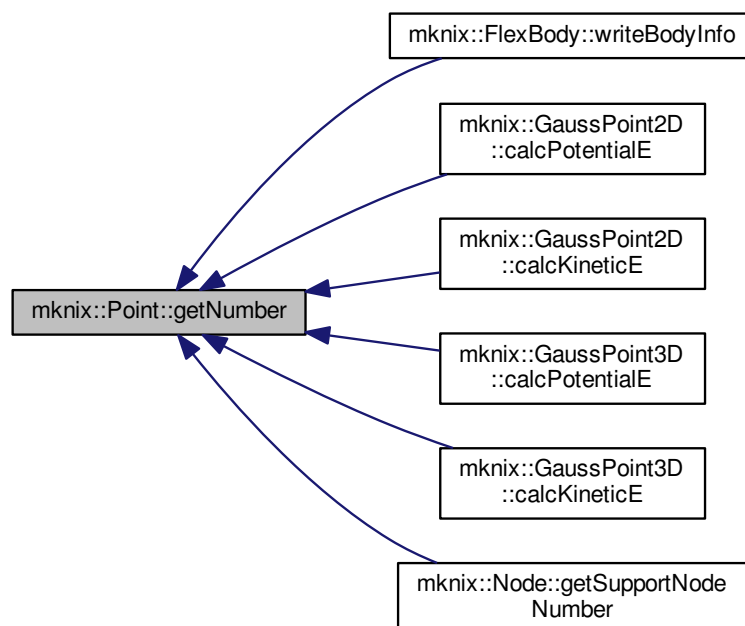
6.44.3.6 `const int& mknix::Point::getDim ( ) const` `[inline]`

Definition at line 92 of file `point.h`.

6.44.3.7 `const int& mknix::Point::getNumber ( ) const` `[inline]`

Definition at line 112 of file `point.h`.

Here is the caller graph for this function:



6.44.3.8 `std::string mknix::Point::getShapeFunType ( )` `[inline]`

Definition at line 128 of file `point.h`.

6.44.3.9 `int mknix::Point::getSupportNodeNumber ( int deriv, int s_node )` `[virtual]`

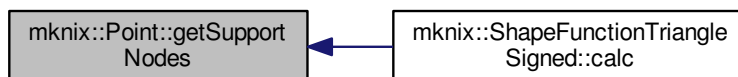
Reimplemented in [mknix::Node](#).

Definition at line 142 of file point.cpp.

6.44.3.10 `const std::vector<Node*>& mknix::Point::getSupportNodes ( ) const` `[inline]`

Definition at line 169 of file point.h.

Here is the caller graph for this function:



6.44.3.11 `virtual size_t mknix::Point::getSupportSize ( int deriv = 0 )` `[inline]`, `[virtual]`

Reimplemented in [mknix::Node](#).

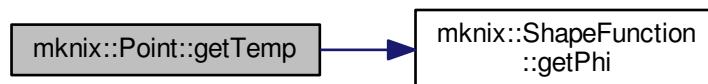
Definition at line 133 of file point.h.

6.44.3.12 `double mknix::Point::getTemp ( ) const` `[virtual]`

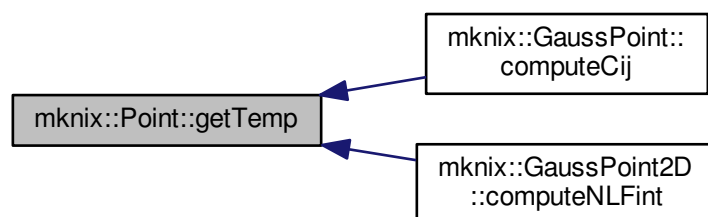
Reimplemented in [mknix::Node](#).

Definition at line 117 of file point.cpp.

Here is the call graph for this function:



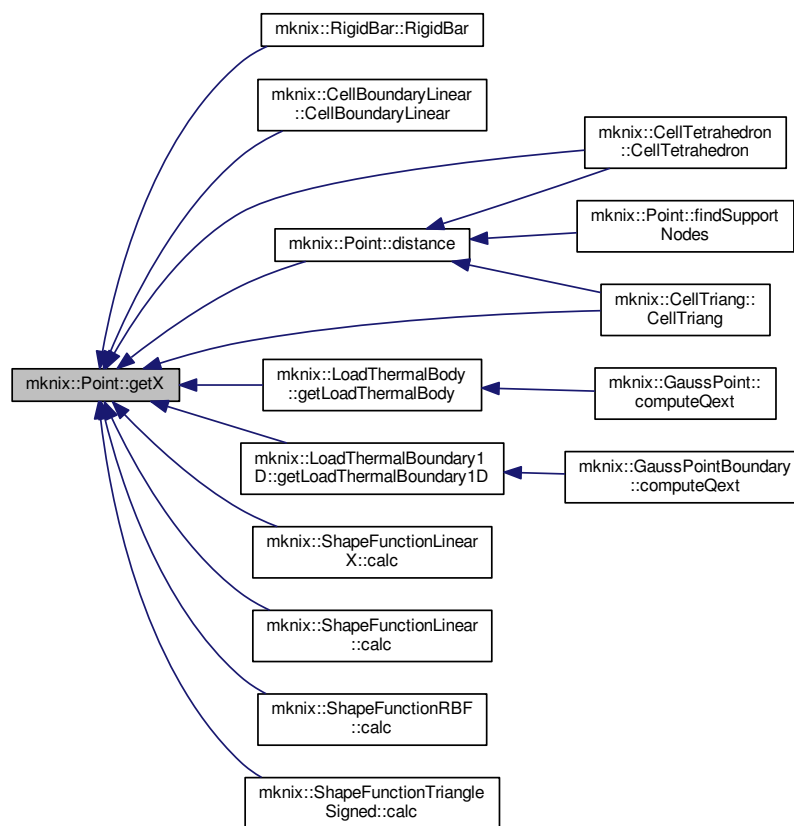
Here is the caller graph for this function:



#### 6.44.3.13 `const double& mknix::Point::getX ( ) const [inline]`

Definition at line 97 of file point.h.

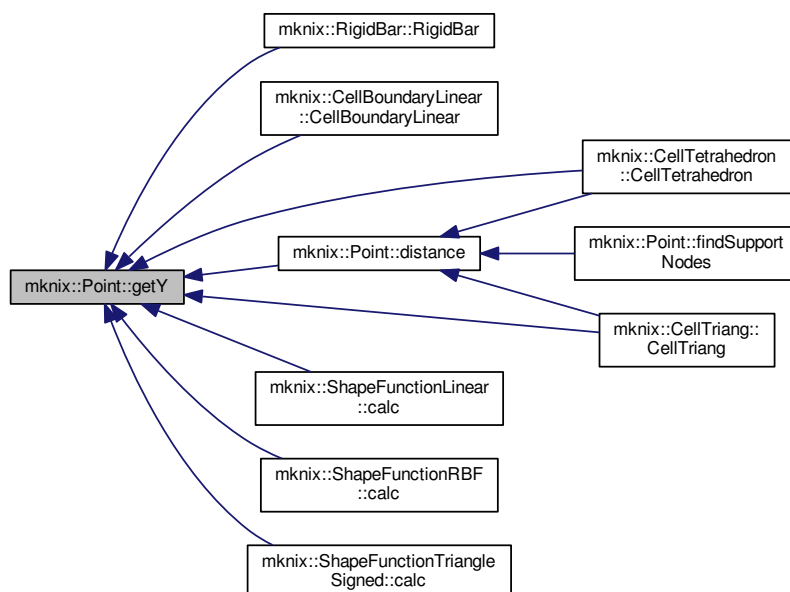
Here is the caller graph for this function:



#### 6.44.3.14 `const double& mknix::Point::getY ( ) const [inline]`

Definition at line 102 of file point.h.

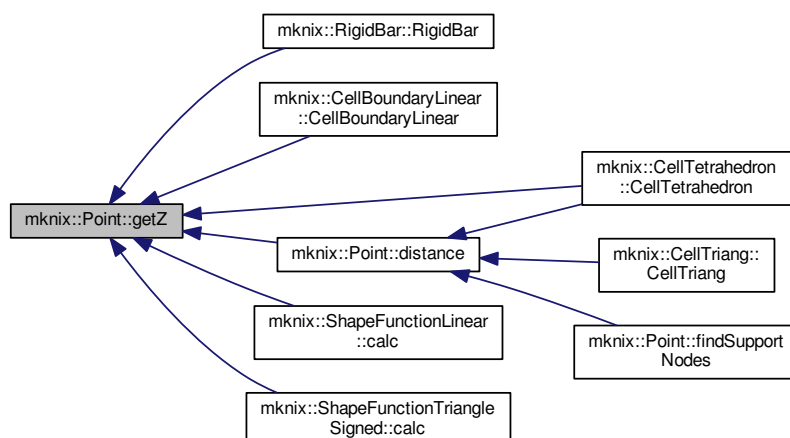
Here is the caller graph for this function:



#### 6.44.3.15 `const double& mknix::Point::getZ ( ) const [inline]`

Definition at line 107 of file point.h.

Here is the caller graph for this function:

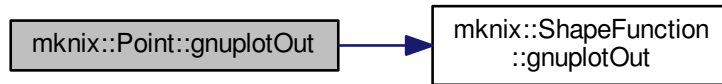


#### 6.44.3.16 `void mknix::Point::gnuplotOut ( std::ofstream & gpdata )`

Definition at line 306 of file point.cpp.



Here is the call graph for this function:



6.44.3.17 `void mknix::Point::setAlphai ( double alphai_in ) [inline]`

Definition at line 145 of file `point.h`.

6.44.3.18 `void mknix::Point::setDc ( double dc_in ) [inline]`

Definition at line 140 of file `point.h`.

6.44.3.19 `void mknix::Point::setJacobian ( double jac_in ) [inline]`

Definition at line 162 of file `point.h`.

6.44.3.20 `void mknix::Point::setShapeFunType ( std::string type_in ) [inline]`

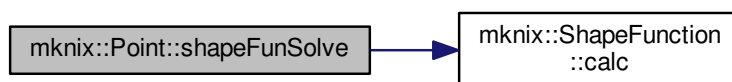
Definition at line 123 of file `point.h`.

6.44.3.21 `void mknix::Point::shapeFunSolve ( std::string type_in, double q_in ) [virtual]`

Reimplemented in [mknix::GaussPointBoundary](#), [mknix::GaussPoint](#), [mknix::GaussPoint2D](#), and [mknix::GaussPoint3D](#).

Definition at line 264 of file `point.cpp`.

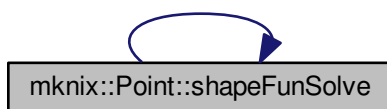
Here is the call graph for this function:



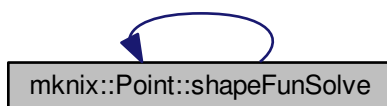
6.44.3.22 `void mknix::Point::shapeFunSolve ( double q_in ) [inline]`

Definition at line 160 of file `point.h`.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.44.4 Friends And Related Function Documentation

##### 6.44.4.1 friend class `GaussPoint` [friend]

Definition at line 59 of file `point.h`.

##### 6.44.4.2 friend class `ShapeFunction` [friend]

Definition at line 61 of file `point.h`.

##### 6.44.4.3 friend class `ShapeFunctionLinear` [friend]

Definition at line 71 of file `point.h`.

##### 6.44.4.4 friend class `ShapeFunctionLinearX` [friend]

Definition at line 73 of file `point.h`.

##### 6.44.4.5 friend class `ShapeFunctionMLS` [friend]

Definition at line 65 of file `point.h`.

##### 6.44.4.6 friend class `ShapeFunctionRBF` [friend]

Definition at line 63 of file `point.h`.

##### 6.44.4.7 friend class `ShapeFunctionTetrahedron` [friend]

Definition at line 67 of file `point.h`.

#### 6.44.4.8 friend class ShapeFunctionTriangle [friend]

Definition at line 69 of file point.h.

### 6.44.5 Member Data Documentation

#### 6.44.5.1 double mknix::Point::alpha [protected]

Definition at line 178 of file point.h.

#### 6.44.5.2 double mknix::Point::dc [protected]

Definition at line 179 of file point.h.

#### 6.44.5.3 int mknix::Point::dim [protected]

Definition at line 175 of file point.h.

#### 6.44.5.4 double mknix::Point::jacobian [protected]

Definition at line 184 of file point.h.

#### 6.44.5.5 int mknix::Point::num [protected]

Definition at line 176 of file point.h.

#### 6.44.5.6 ShapeFunction\* mknix::Point::shapeFun [protected]

Definition at line 181 of file point.h.

#### 6.44.5.7 std::string mknix::Point::shapeFunType [protected]

Definition at line 180 of file point.h.

#### 6.44.5.8 std::vector<Node\*> mknix::Point::supportNodes [protected]

Definition at line 182 of file point.h.

#### 6.44.5.9 size\_t mknix::Point::supportNodesSize [protected]

Definition at line 183 of file point.h.

#### 6.44.5.10 double mknix::Point::X [protected]

Definition at line 177 of file point.h.

#### 6.44.5.11 double mknix::Point::Y [protected]

Definition at line 177 of file point.h.

#### 6.44.5.12 double mknix::Point::Z [protected]

Definition at line 177 of file point.h.

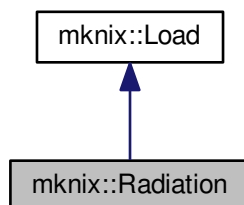
The documentation for this class was generated from the following files:

- [point.h](#)
- [point.cpp](#)

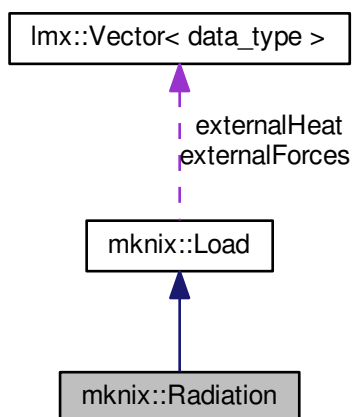
## 6.45 mknix::Radiation Class Reference

```
#include <loadradiation.h>
```

Inheritance diagram for mknix::Radiation:



Collaboration diagram for mknix::Radiation:



### Public Member Functions

- [Radiation](#) ()
- [~Radiation](#) ()
- void [addVoxel](#) (double, double, double, double)
- void [outputToFile](#) (std::ofstream \*)

### Additional Inherited Members

#### 6.45.1 Detailed Description

Author

AUTHORS <MAILS>

Definition at line 32 of file loadradiation.h.

#### 6.45.2 Constructor & Destructor Documentation

##### 6.45.2.1 mknix::Radiation::Radiation ( )

Definition at line 24 of file loadradiation.cpp.

##### 6.45.2.2 mknix::Radiation::~~Radiation ( )

Definition at line 29 of file loadradiation.cpp.

#### 6.45.3 Member Function Documentation

##### 6.45.3.1 void mknix::Radiation::addVoxel ( double *x\_in*, double *y\_in*, double *z\_in*, double *value\_in* )

Definition at line 34 of file loadradiation.cpp.

##### 6.45.3.2 void mknix::Radiation::outputToFile ( std::ofstream \* *outFile* ) [virtual]

Implements [mknix::Load](#).

Definition at line 40 of file loadradiation.cpp.

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

- [loadradiation.h](#)
- [loadradiation.cpp](#)

## 6.46 mknix::Reader Class Reference

```
#include <reader.h>
```

#### Public Member Functions

- [Reader](#) ()
- [Reader](#) ([Simulation](#) \*)
- [~Reader](#) ()
- void [inputFromFile](#) (const std::string &fileIn)

#### 6.46.1 Detailed Description

Author

AUTHORS <MAILS>

Definition at line 43 of file reader.h.

#### 6.46.2 Constructor & Destructor Documentation

##### 6.46.2.1 mknix::Reader::Reader ( )

Definition at line 53 of file reader.cpp.

#### 6.46.2.2 `mknix::Reader::Reader ( Simulation * simulation_in )`

Definition at line 60 of file reader.cpp.

#### 6.46.2.3 `mknix::Reader::~Reader ( )`

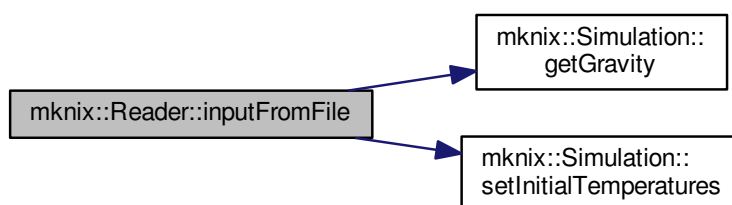
Definition at line 68 of file reader.cpp.

### 6.46.3 Member Function Documentation

#### 6.46.3.1 `void mknix::Reader::inputFromFile ( const std::string & fileName )`

Definition at line 75 of file reader.cpp.

Here is the call graph for this function:



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

- [reader.h](#)
- [reader.cpp](#)

## 6.47 `mknix::ReaderConstraints` Class Reference

```
#include <readerconstraints.h>
```

### Public Member Functions

- [ReaderConstraints](#) ()
- [ReaderConstraints](#) ([Simulation](#) \*, [std::ofstream](#) &, [std::ifstream](#) &)
- [~ReaderConstraints](#) ()
- void [readConstraints](#) ([System](#) \*)

#### 6.47.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 37 of file readerconstraints.h.

## 6.47.2 Constructor &amp; Destructor Documentation

## 6.47.2.1 mknix::ReaderConstraints::ReaderConstraints ( )

Definition at line 37 of file readerconstraints.cpp.

6.47.2.2 mknix::ReaderConstraints::ReaderConstraints ( Simulation \* *simulation\_in*, std::ofstream & *output\_in*, std::ifstream & *input\_in* )

Definition at line 46 of file readerconstraints.cpp.

## 6.47.2.3 mknix::ReaderConstraints::~~ReaderConstraints ( )

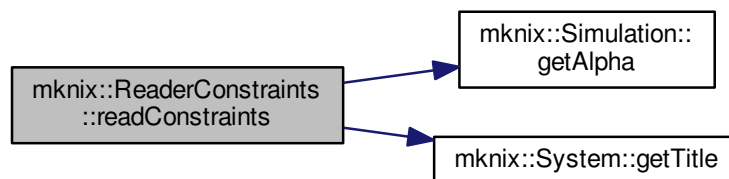
Definition at line 59 of file readerconstraints.cpp.

## 6.47.3 Member Function Documentation

6.47.3.1 void mknix::ReaderConstraints::readConstraints ( System \* *system\_in* )

Definition at line 66 of file readerconstraints.cpp.

Here is the call graph for this function:



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

- [readerconstraints.h](#)
- [readerconstraints.cpp](#)

## 6.48 mknix::ReaderFlex Class Reference

```
#include <readerflex.h>
```

## Public Member Functions

- [ReaderFlex](#) ( )
- [ReaderFlex](#) (Simulation \*, std::ofstream &, std::ifstream &)
- [~ReaderFlex](#) ( )
- void [readFlexBodies](#) (System \*)

## 6.48.1 Detailed Description

**Author**

AUTHORS <MAILS>

Definition at line 34 of file readerflex.h.

**6.48.2 Constructor & Destructor Documentation****6.48.2.1 mknix::ReaderFlex::ReaderFlex ( )**

Definition at line 35 of file readerflex.cpp.

**6.48.2.2 mknix::ReaderFlex::ReaderFlex ( Simulation \* *simulation\_in*, std::ofstream & *output\_in*, std::ifstream & *input\_in* )**

Definition at line 42 of file readerflex.cpp.

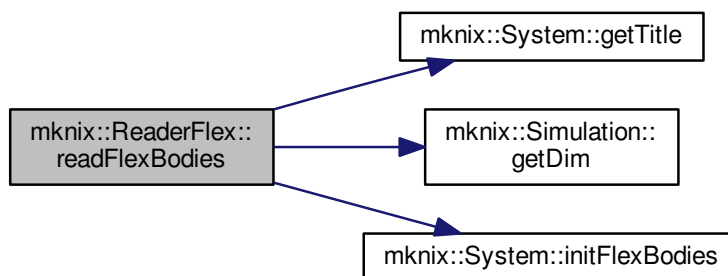
**6.48.2.3 mknix::ReaderFlex::~~ReaderFlex ( )**

Definition at line 53 of file readerflex.cpp.

**6.48.3 Member Function Documentation****6.48.3.1 void mknix::ReaderFlex::readFlexBodies ( System \* *system\_in* )**

Definition at line 61 of file readerflex.cpp.

Here is the call graph for this function:



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

- [readerflex.h](#)
- [readerflex.cpp](#)

**6.49 mknix::ReaderRigid Class Reference**

```
#include <readerrigid.h>
```

**Public Member Functions**

- [ReaderRigid \( \)](#)



- [ReaderRigid](#) ([Simulation](#) \*, [std::ofstream](#) &, [std::ifstream](#) &)
- [~ReaderRigid](#) ()
- void [readRigidBodies](#) ([System](#) \*)

#### 6.49.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 36 of file readerrigid.h.

#### 6.49.2 Constructor & Destructor Documentation

##### 6.49.2.1 mknix::ReaderRigid::ReaderRigid ( )

Definition at line 34 of file readerrigid.cpp.

##### 6.49.2.2 mknix::ReaderRigid::ReaderRigid ( [Simulation](#) \* *simulation\_in*, [std::ofstream](#) & *output\_in*, [std::ifstream](#) & *input\_in* )

Definition at line 41 of file readerrigid.cpp.

##### 6.49.2.3 mknix::ReaderRigid::~ReaderRigid ( )

Definition at line 51 of file readerrigid.cpp.

#### 6.49.3 Member Function Documentation

##### 6.49.3.1 void mknix::ReaderRigid::readRigidBodies ( [System](#) \* *system\_in* )

Definition at line 58 of file readerrigid.cpp.

Here is the call graph for this function:



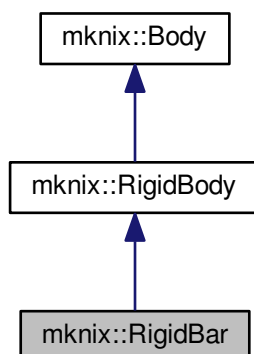
The documentation for this class was generated from the following files:

- [readerrigid.h](#)
- [readerrigid.cpp](#)

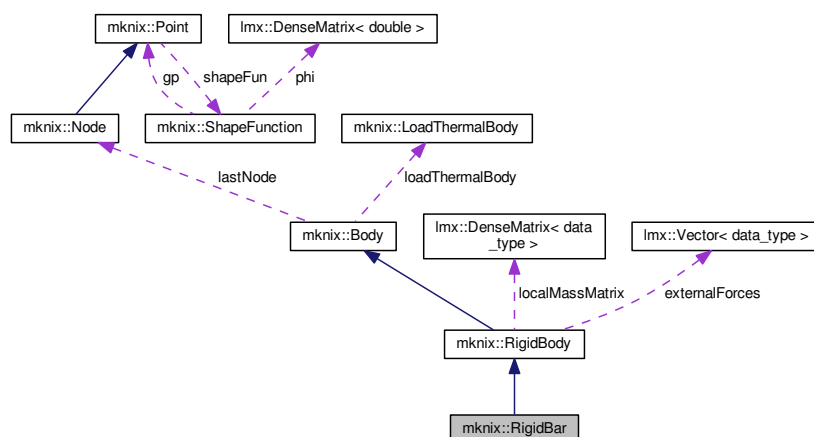
## 6.50 mknix::RigidBar Class Reference

```
#include <bodyrigid1D.h>
```

Inheritance diagram for `mknix::RigidBody`:



Collaboration diagram for `mknix::RigidBody`:



### Public Member Functions

- `RigidBody ()`
- `RigidBody (std::string, Node *, Node *, double)`
- `~RigidBody ()`
- `std::string getType ()`
- `void setInertia (double, int)`
- `void setPosition (std::vector< double > &)`
- `void calcMassMatrix ()`
- `void calcExternalForces ()`
- `void addNode (Node *)`
- `void writeBoundaryNodes (std::vector< Node * > &)`
- `void writeBoundaryConnectivity (std::vector< std::vector< Node * > > &)`

## Additional Inherited Members

### 6.50.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 31 of file bodyrigid1D.h.

### 6.50.2 Constructor & Destructor Documentation

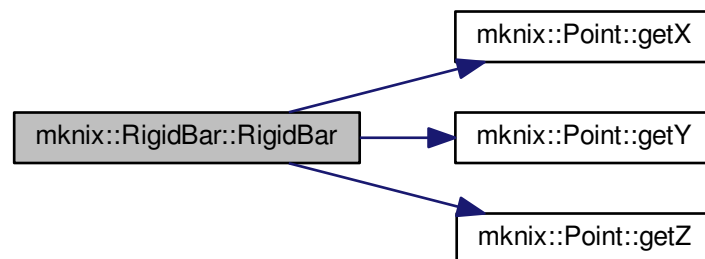
#### 6.50.2.1 mknix::RigidBar::RigidBar ( )

Definition at line 26 of file bodyrigid1D.cpp.

#### 6.50.2.2 mknix::RigidBar::RigidBar ( std::string *title\_in*, Node \* *nodeA\_in*, Node \* *nodeB\_in*, double *mass\_in* )

Definition at line 32 of file bodyrigid1D.cpp.

Here is the call graph for this function:



#### 6.50.2.3 mknix::RigidBar::~~RigidBar ( )

Definition at line 55 of file bodyrigid1D.cpp.

### 6.50.3 Member Function Documentation

#### 6.50.3.1 void mknix::RigidBar::addNode ( Node \* *node\_in* ) [virtual]

Reimplemented from [mknix::Body](#).

Definition at line 110 of file bodyrigid1D.cpp.

Here is the call graph for this function:

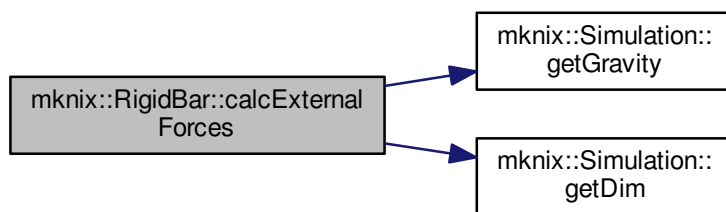


#### 6.50.3.2 `void mknix::RigidBody::calcExternalForces ( ) [virtual]`

Implements [mnix::Body](#).

Definition at line 95 of file `bodyrigid1D.cpp`.

Here is the call graph for this function:



#### 6.50.3.3 `void mknix::RigidBody::calcMassMatrix ( ) [virtual]`

Implements [mnix::Body](#).

Definition at line 75 of file `bodyrigid1D.cpp`.

Here is the call graph for this function:



#### 6.50.3.4 `std::string mknix::RigidBody::getType ( ) [inline],[virtual]`

Implements [mnix::Body](#).

Definition at line 40 of file `bodyrigid1D.h`.

6.50.3.5 void mknix::RigidBar::setInertia ( double *inertia\_in*, int *axis* ) [virtual]

Implements [mknix::RigidBody](#).

Definition at line 59 of file bodyrigid1D.cpp.

6.50.3.6 void mknix::RigidBar::setPosition ( std::vector< double > & *position* ) [virtual]

Implements [mknix::RigidBody](#).

Definition at line 66 of file bodyrigid1D.cpp.

6.50.3.7 void mknix::RigidBar::writeBoundaryConnectivity ( std::vector< std::vector< Node \* > > & *connectivity\_nodes* )

Definition at line 135 of file bodyrigid1D.cpp.

6.50.3.8 void mknix::RigidBar::writeBoundaryNodes ( std::vector< Node \* > & *boundary\_nodes* )

Definition at line 127 of file bodyrigid1D.cpp.

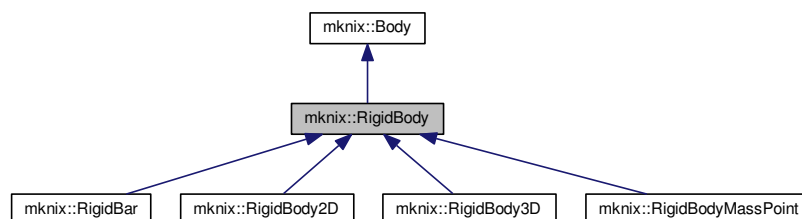
The documentation for this class was generated from the following files:

- [bodyrigid1D.h](#)
- [bodyrigid1D.cpp](#)

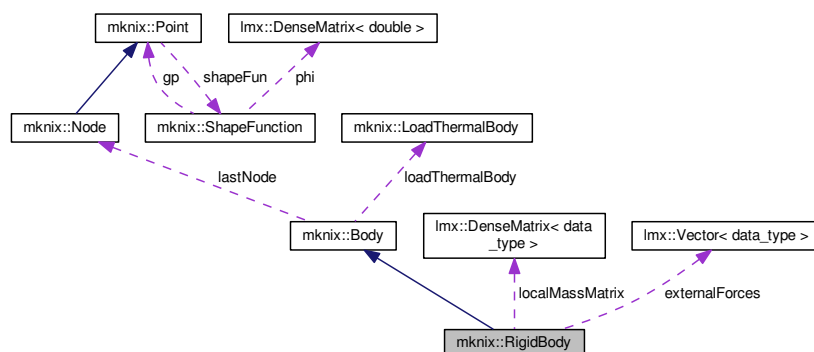
## 6.51 mknix::RigidBody Class Reference

```
#include <bodyrigid.h>
```

Inheritance diagram for mknix::RigidBody:



Collaboration diagram for `mknix::RigidBody`:



### Public Member Functions

- `RigidBody ()`
- `RigidBody (std::string)`
- `virtual ~RigidBody ()`
- `void assembleMassMatrix (Imx::Matrix< data_type > &)`
- `void assembleExternalForces (Imx::Vector< data_type > &)`
- `void setMass (double mass_in)`
- `void setDensityFactor (double density_in)`
- `virtual void setInertia (double, int)=0`
- `virtual void setPosition (std::vector< double > &)=0`
- `void setOutput (std::string)`
- `Node * getNode (int)`
- `void outputStep (const Imx::Vector< data_type > &, const Imx::Vector< data_type > &)`
- `void outputStep (const Imx::Vector< data_type > &)`
- `void outputToFile (std::ofstream *)`  
*Streams the data stored during the analysis to a file.*
- `void writeBodyInfo (std::ofstream *)`
- `virtual void writeBoundaryNodes (std::vector< Point * > &)`
- `virtual void writeBoundaryConnectivity (std::vector< std::vector< Point * > > &)`

### Protected Attributes

- `bool computeEnergy`
- `int dim`
- `double mass`
- `double densityFactor`
- `std::vector< Node * > frameNodes`
- `std::vector< Imx::Vector< data_type > * > domainConf`
- `Imx::Vector< data_type > externalForces`
- `Imx::DenseMatrix< data_type > localMassMatrix`
- `std::vector< Imx::Vector< data_type > * > energy`

## 6.51.1 Detailed Description

## Author

AUTHORS &lt;MAILS&gt;

Definition at line 33 of file bodyrigid.h.

## 6.51.2 Constructor &amp; Destructor Documentation

## 6.51.2.1 mknix::RigidBody::RigidBody ( )

Definition at line 26 of file bodyrigid.cpp.

6.51.2.2 mknix::RigidBody::RigidBody ( std::string *title\_in* )

Definition at line 35 of file bodyrigid.cpp.

## 6.51.2.3 mknix::RigidBody::~~RigidBody ( ) [virtual]

Definition at line 44 of file bodyrigid.cpp.

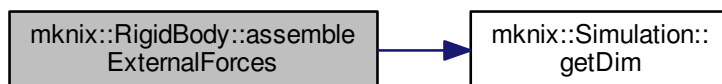
## 6.51.3 Member Function Documentation

6.51.3.1 void mknix::RigidBody::assembleExternalForces ( Imx::Vector< data\_type > & *globalExternalForces* ) [virtual]

Implements [mnix::Body](#).

Definition at line 71 of file bodyrigid.cpp.

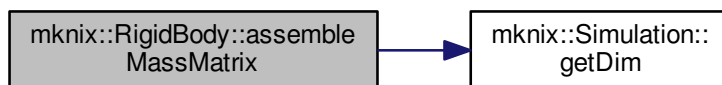
Here is the call graph for this function:

6.51.3.2 void mknix::RigidBody::assembleMassMatrix ( Imx::Matrix< data\_type > & *globalMass* ) [virtual]

Implements [mnix::Body](#).

Definition at line 49 of file bodyrigid.cpp.

Here is the call graph for this function:



#### 6.51.3.3 `Node * mknix::RigidBody::getNode ( int node_number ) [virtual]`

Reimplemented from [mknix::Body](#).

Definition at line 97 of file `bodyrigid.cpp`.

#### 6.51.3.4 `void mknix::RigidBody::outputStep ( const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot ) [virtual]`

Implements [mknix::Body](#).

Definition at line 122 of file `bodyrigid.cpp`.

Here is the call graph for this function:



#### 6.51.3.5 `void mknix::RigidBody::outputStep ( const Imx::Vector< data_type > & q ) [virtual]`

Implements [mknix::Body](#).

Definition at line 174 of file `bodyrigid.cpp`.

Here is the call graph for this function:



#### 6.51.3.6 `void mknix::RigidBody::outputToFile ( std::ofstream * outFile ) [virtual]`

Streams the data stored during the analysis to a file.



## Parameters

<i>outFile</i>	Output files
----------------	--------------

## Returns

void

Reimplemented from [mknix::Body](#).

Definition at line 210 of file bodyrigid.cpp.

Here is the call graph for this function:

6.51.3.7 void mknix::RigidBody::setDensityFactor ( double *density\_in* ) [inline]

Definition at line 50 of file bodyrigid.h.

6.51.3.8 virtual void mknix::RigidBody::setInertia ( double , int ) [pure virtual]

Implemented in [mknix::RigidBar](#), [mknix::RigidBody2D](#), [mknix::RigidBody3D](#), and [mknix::RigidBodyMassPoint](#).6.51.3.9 void mknix::RigidBody::setMass ( double *mass\_in* ) [inline]

Definition at line 47 of file bodyrigid.h.

6.51.3.10 void mknix::RigidBody::setOutput ( std::string *outputType\_in* ) [virtual]Implements [mknix::Body](#).

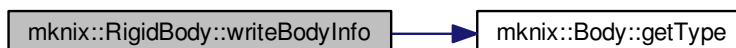
Definition at line 92 of file bodyrigid.cpp.

6.51.3.11 virtual void mknix::RigidBody::setPosition ( std::vector&lt; double &gt; &amp; ) [pure virtual]

Implemented in [mknix::RigidBar](#), [mknix::RigidBody2D](#), [mknix::RigidBody3D](#), and [mknix::RigidBodyMassPoint](#).6.51.3.12 void mknix::RigidBody::writeBodyInfo ( std::ofstream \* *outFile* ) [virtual]Implements [mknix::Body](#).

Definition at line 243 of file bodyrigid.cpp.

Here is the call graph for this function:



**6.51.3.13** `void mknix::RigidBody::writeBoundaryConnectivity ( std::vector< std::vector< Point * > > & connectivity_nodes ) [virtual]`

Implements [mknix::Body](#).

Definition at line 269 of file `bodyrigid.cpp`.

**6.51.3.14** `void mknix::RigidBody::writeBoundaryNodes ( std::vector< Point * > & boundary_nodes ) [virtual]`

Implements [mknix::Body](#).

Definition at line 258 of file `bodyrigid.cpp`.

#### 6.51.4 Member Data Documentation

**6.51.4.1** `bool mknix::RigidBody::computeEnergy [protected]`

Definition at line 74 of file `bodyrigid.h`.

**6.51.4.2** `double mknix::RigidBody::densityFactor [protected]`

Definition at line 76 of file `bodyrigid.h`.

**6.51.4.3** `int mknix::RigidBody::dim [protected]`

Definition at line 75 of file `bodyrigid.h`.

**6.51.4.4** `std::vector< Imx::Vector<data_type>* > mknix::RigidBody::domainConf [protected]`

Definition at line 78 of file `bodyrigid.h`.

**6.51.4.5** `std::vector< Imx::Vector<data_type>* > mknix::RigidBody::energy [protected]`

Definition at line 81 of file `bodyrigid.h`.

**6.51.4.6** `Imx::Vector<data_type> mknix::RigidBody::externalForces [protected]`

Definition at line 79 of file `bodyrigid.h`.

**6.51.4.7** `std::vector<Node*> mknix::RigidBody::frameNodes [protected]`

Definition at line 77 of file `bodyrigid.h`.

**6.51.4.8** `Imx::DenseMatrix< data_type > mknix::RigidBody::localMassMatrix [protected]`

Definition at line 80 of file `bodyrigid.h`.

**6.51.4.9** `double mknix::RigidBody::mass [protected]`

Definition at line 76 of file `bodyrigid.h`.

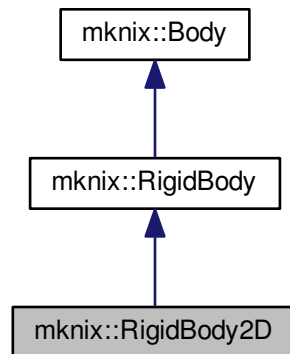
The documentation for this class was generated from the following files:

- [bodyrigid.h](#)
- [bodyrigid.cpp](#)

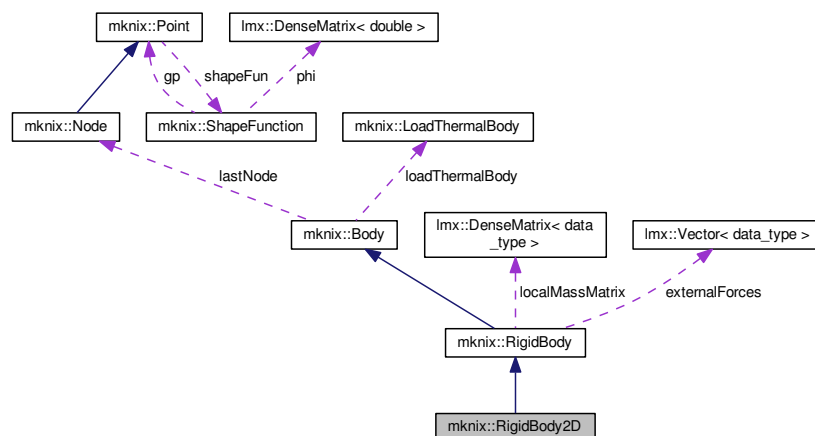
## 6.52 mknix::RigidBody2D Class Reference

```
#include <bodyrigid2D.h>
```

Inheritance diagram for mknix::RigidBody2D:



Collaboration diagram for mknix::RigidBody2D:



### Public Member Functions

- [RigidBody2D](#) ()
- [RigidBody2D](#) (std::string, [Node](#) \*, [Node](#) \*, [Node](#) \*)
- [~RigidBody2D](#) ()
- std::string [getType](#) ()
- void [setInertia](#) (double, int)
- void [setPosition](#) (std::vector< double > &)
- void [calcMassMatrix](#) ()
- void [calcExternalForces](#) ()
- void [addNode](#) ([Node](#) \*)

## Additional Inherited Members

### 6.52.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 31 of file bodyrigid2D.h.

### 6.52.2 Constructor & Destructor Documentation

#### 6.52.2.1 `mknix::RigidBody2D::RigidBody2D ( )`

Definition at line 26 of file bodyrigid2D.cpp.

#### 6.52.2.2 `mknix::RigidBody2D::RigidBody2D ( std::string title_in, Node * nodeA_in, Node * nodeB_in, Node * nodeC_in )`

Definition at line 34 of file bodyrigid2D.cpp.

#### 6.52.2.3 `mknix::RigidBody2D::~~RigidBody2D ( )`

Definition at line 58 of file bodyrigid2D.cpp.

### 6.52.3 Member Function Documentation

#### 6.52.3.1 `void mknix::RigidBody2D::addNode ( Node * node_in ) [virtual]`

Reimplemented from [mknix::Body](#).

Definition at line 132 of file bodyrigid2D.cpp.

Here is the call graph for this function:

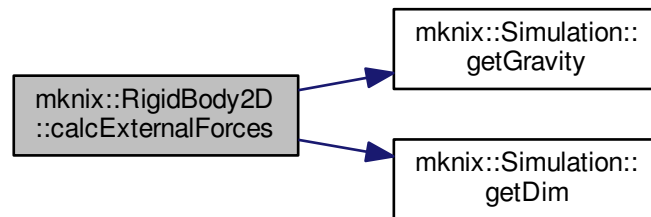


#### 6.52.3.2 `void mknix::RigidBody2D::calcExternalForces ( ) [virtual]`

Implements [mknix::Body](#).

Definition at line 124 of file bodyrigid2D.cpp.

Here is the call graph for this function:

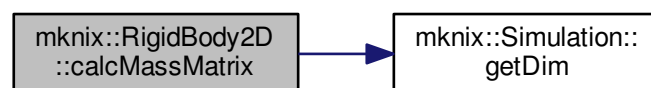


#### 6.52.3.3 void mknix::RigidBody2D::calcMassMatrix ( ) [virtual]

Implements [mknix::Body](#).

Definition at line 81 of file `bodyrigid2D.cpp`.

Here is the call graph for this function:



#### 6.52.3.4 std::string mknix::RigidBody2D::getType ( ) [inline],[virtual]

Implements [mknix::Body](#).

Definition at line 40 of file `bodyrigid2D.h`.

#### 6.52.3.5 void mknix::RigidBody2D::setInertia ( double *inertia\_in*, int *axis* ) [virtual]

Implements [mknix::RigidBody](#).

Definition at line 62 of file `bodyrigid2D.cpp`.

#### 6.52.3.6 void mknix::RigidBody2D::setPosition ( std::vector< double > & *position* ) [virtual]

Implements [mknix::RigidBody](#).

Definition at line 69 of file `bodyrigid2D.cpp`.

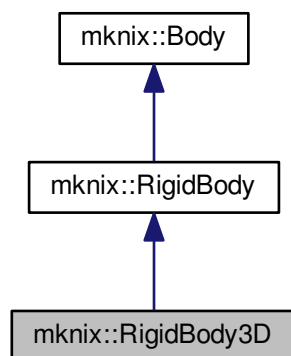
The documentation for this class was generated from the following files:

- [bodyrigid2D.h](#)
- [bodyrigid2D.cpp](#)

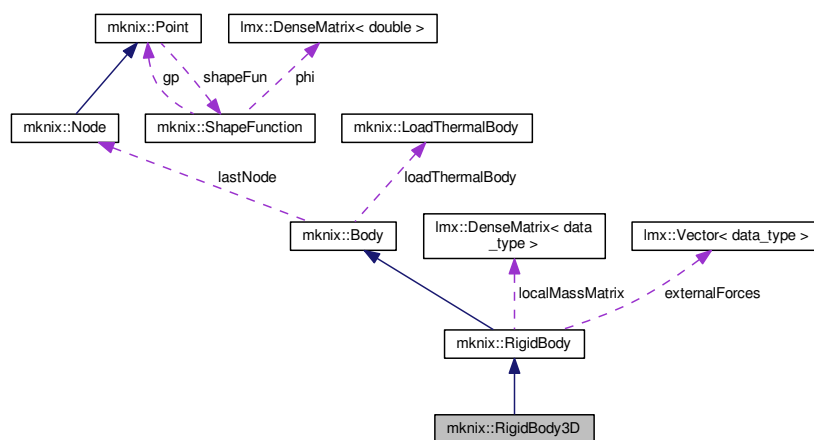
### 6.53 mknix::RigidBody3D Class Reference

```
#include <bodyrigid3D.h>
```

Inheritance diagram for mknix::RigidBody3D:



Collaboration diagram for mknix::RigidBody3D:



#### Public Member Functions

- [RigidBody3D](#) ()
- [RigidBody3D](#) (std::string, [Node](#) \*, [Node](#) \*, [Node](#) \*, [Node](#) \*)
- [~RigidBody3D](#) ()
- std::string [getType](#) ()
- void [setInertia](#) (double, int)
- void [setPosition](#) (std::vector< double > &)
- void [calcMassMatrix](#) ()
- void [calcExternalForces](#) ()
- void [addNode](#) ([Node](#) \*)

## Additional Inherited Members

### 6.53.1 Detailed Description

Definition at line 28 of file bodyrigid3D.h.

### 6.53.2 Constructor & Destructor Documentation

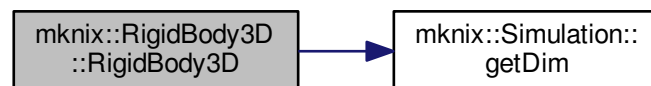
#### 6.53.2.1 mknix::RigidBody3D::RigidBody3D ( )

Definition at line 32 of file bodyrigid3D.cpp.

#### 6.53.2.2 mknix::RigidBody3D::RigidBody3D ( std::string *title\_in*, Node \* *node0\_in*, Node \* *node1\_in*, Node \* *node2\_in*, Node \* *node3\_in* )

Definition at line 36 of file bodyrigid3D.cpp.

Here is the call graph for this function:



#### 6.53.2.3 mknix::RigidBody3D::~~RigidBody3D ( )

Definition at line 53 of file bodyrigid3D.cpp.

### 6.53.3 Member Function Documentation

#### 6.53.3.1 void mknix::RigidBody3D::addNode ( Node \* *node\_in* ) [virtual]

Reimplemented from [mknix::Body](#).

Definition at line 174 of file bodyrigid3D.cpp.

Here is the call graph for this function:



**6.53.3.2** `void mknix::RigidBody3D::calcExternalForces ( ) [virtual]`

Implements [mknix::Body](#).

Definition at line 164 of file `bodyrigid3D.cpp`.

Here is the call graph for this function:



**6.53.3.3** `void mknix::RigidBody3D::calcMassMatrix ( ) [virtual]`

Implements [mknix::Body](#).

Definition at line 88 of file `bodyrigid3D.cpp`.

**6.53.3.4** `std::string mknix::RigidBody3D::getType ( ) [inline],[virtual]`

Implements [mknix::Body](#).

Definition at line 38 of file `bodyrigid3D.h`.

**6.53.3.5** `void mknix::RigidBody3D::setInertia ( double inertia_in, int axis ) [virtual]`

Implements [mknix::RigidBody](#).

Definition at line 57 of file `bodyrigid3D.cpp`.

**6.53.3.6** `void mknix::RigidBody3D::setPosition ( std::vector< double > & position ) [virtual]`

Implements [mknix::RigidBody](#).

Definition at line 68 of file `bodyrigid3D.cpp`.

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

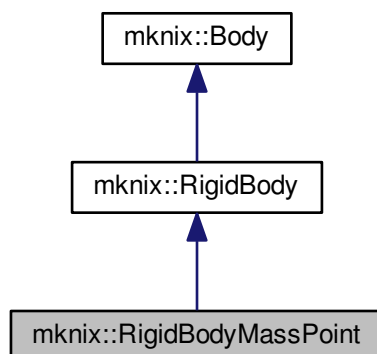
- [bodyrigid3D.h](#)
- [bodyrigid3D.cpp](#)

## 6.54 mknix::RigidBodyMassPoint Class Reference

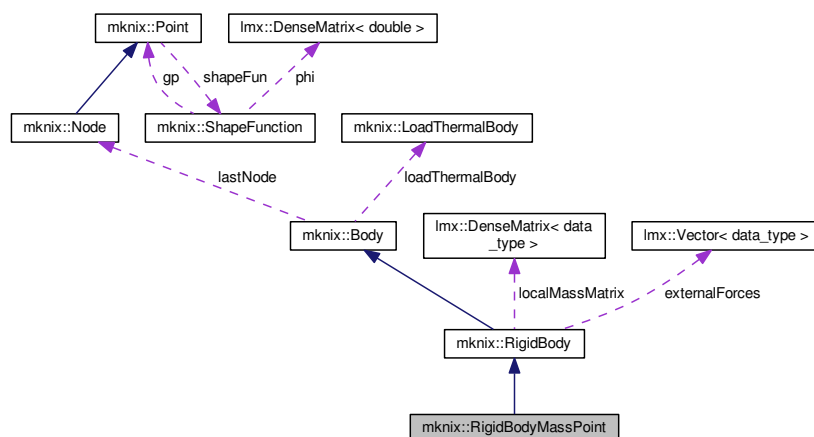
```
#include <bodyrigid0D.h>
```



Inheritance diagram for mknix::RigidBodyMassPoint:



Collaboration diagram for mknix::RigidBodyMassPoint:



#### Public Member Functions

- [RigidBodyMassPoint](#) ()
- [RigidBodyMassPoint](#) (std::string, [Node](#) \*, double)
- [~RigidBodyMassPoint](#) ()
- void [setInertia](#) (double, int)
- void [setPosition](#) (std::vector< double > &)
- void [calcMassMatrix](#) ()
- void [calcExternalForces](#) ()
- std::string [getType](#) ()

#### Additional Inherited Members

### 6.54.1 Detailed Description

Definition at line 29 of file bodyrigid0D.h.

### 6.54.2 Constructor & Destructor Documentation

#### 6.54.2.1 `mknix::RigidBodyMassPoint::RigidBodyMassPoint ( )`

Definition at line 31 of file bodyrigid0D.cpp.

#### 6.54.2.2 `mknix::RigidBodyMassPoint::RigidBodyMassPoint ( std::string title_in, Node * nodeA_in, double mass_in )`

Definition at line 35 of file bodyrigid0D.cpp.

#### 6.54.2.3 `mknix::RigidBodyMassPoint::~~RigidBodyMassPoint ( )`

Definition at line 47 of file bodyrigid0D.cpp.

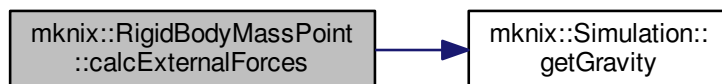
### 6.54.3 Member Function Documentation

#### 6.54.3.1 `void mknix::RigidBodyMassPoint::calcExternalForces ( ) [virtual]`

Implements [mknix::Body](#).

Definition at line 83 of file bodyrigid0D.cpp.

Here is the call graph for this function:



#### 6.54.3.2 `void mknix::RigidBodyMassPoint::calcMassMatrix ( ) [virtual]`

Implements [mknix::Body](#).

Definition at line 63 of file bodyrigid0D.cpp.

#### 6.54.3.3 `std::string mknix::RigidBodyMassPoint::getType ( ) [inline], [virtual]`

Implements [mknix::Body](#).

Definition at line 48 of file bodyrigid0D.h.

#### 6.54.3.4 `void mknix::RigidBodyMassPoint::setInertia ( double inertia_in, int axis ) [virtual]`

Implements [mknix::RigidBody](#).

Definition at line 51 of file bodyrigid0D.cpp.

#### 6.54.3.5 `void mknix::RigidBodyMassPoint::setPosition ( std::vector< double > & position ) [virtual]`

Implements [mknix::RigidBody](#).

Definition at line 56 of file bodyrigid0D.cpp.

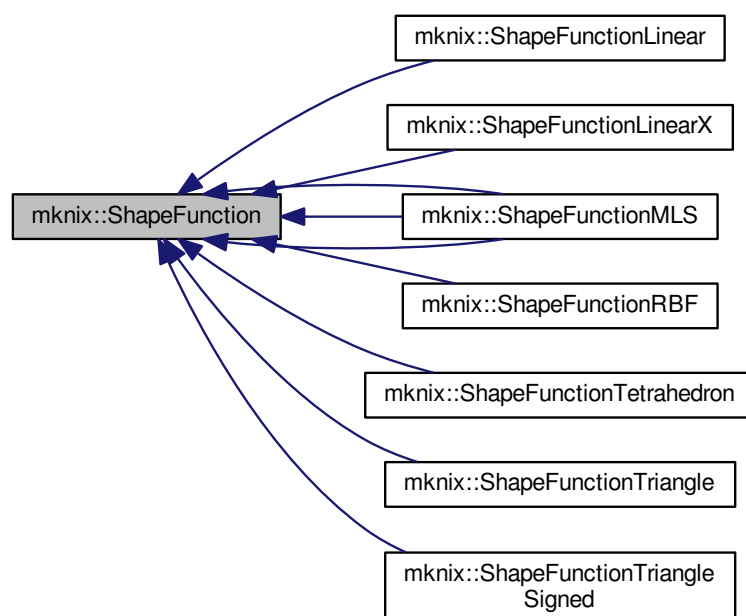
The documentation for this class was generated from the following files:

- [bodyrigid0D.h](#)
- [bodyrigid0D.cpp](#)

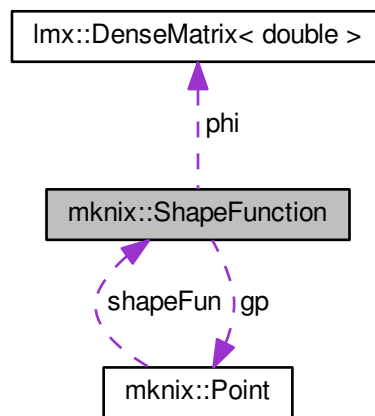
## 6.55 mknix::ShapeFunction Class Reference

```
#include <shapefunction.h>
```

Inheritance diagram for mknix::ShapeFunction:



Collaboration diagram for `mknix::ShapeFunction`:



#### Public Member Functions

- [ShapeFunction](#) ()
- [ShapeFunction](#) (const [ShapeFunction](#) \*)
- [ShapeFunction](#) ([Point](#) \*)
- virtual [~ShapeFunction](#) ()
- virtual void [calc](#) ()=0
- virtual double [getPhi](#) (int i, int j)
- virtual void [setPhi](#) (double value\_in, int i, int j)
- virtual void [outputValues](#) ()
- virtual void [gnuplotOut](#) ()

#### Protected Attributes

- int [dim](#)
- [Imx::DenseMatrix](#)< double > [phi](#)
- [Point](#) \* [gp](#)

#### 6.55.1 Detailed Description

##### Author

Daniel Iglesias

Definition at line 43 of file `shapefunction.h`.

#### 6.55.2 Constructor & Destructor Documentation

##### 6.55.2.1 `mknix::ShapeFunction::ShapeFunction ( )`

Definition at line 9 of file `shapefunction.cpp`.

### 6.55.2.2 mknix::ShapeFunction::ShapeFunction ( const ShapeFunction \* sf\_in )

Definition at line 14 of file shapefunction.cpp.

### 6.55.2.3 mknix::ShapeFunction::ShapeFunction ( Point \* gp\_in )

Definition at line 22 of file shapefunction.cpp.

### 6.55.2.4 mknix::ShapeFunction::~~ShapeFunction ( ) [virtual]

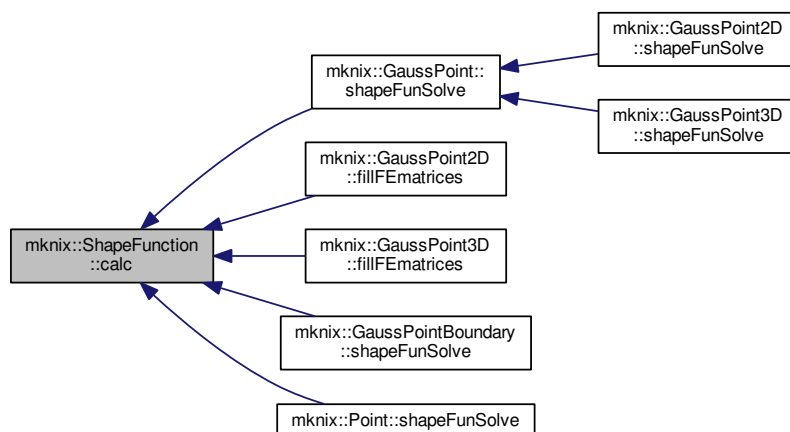
Definition at line 29 of file shapefunction.cpp.

## 6.55.3 Member Function Documentation

### 6.55.3.1 virtual void mknix::ShapeFunction::calc ( ) [pure virtual]

Implemented in [mknix::ShapeFunctionMLS](#), [mknix::ShapeFunctionMLS](#), [mknix::ShapeFunctionMLS](#), [mknix::ShapeFunctionRBF](#), [mknix::ShapeFunctionLinearX](#), [mknix::ShapeFunctionLinear](#), [mknix::ShapeFunctionTetrahedron](#), [mknix::ShapeFunctionTriangle](#), and [mknix::ShapeFunctionTriangleSigned](#).

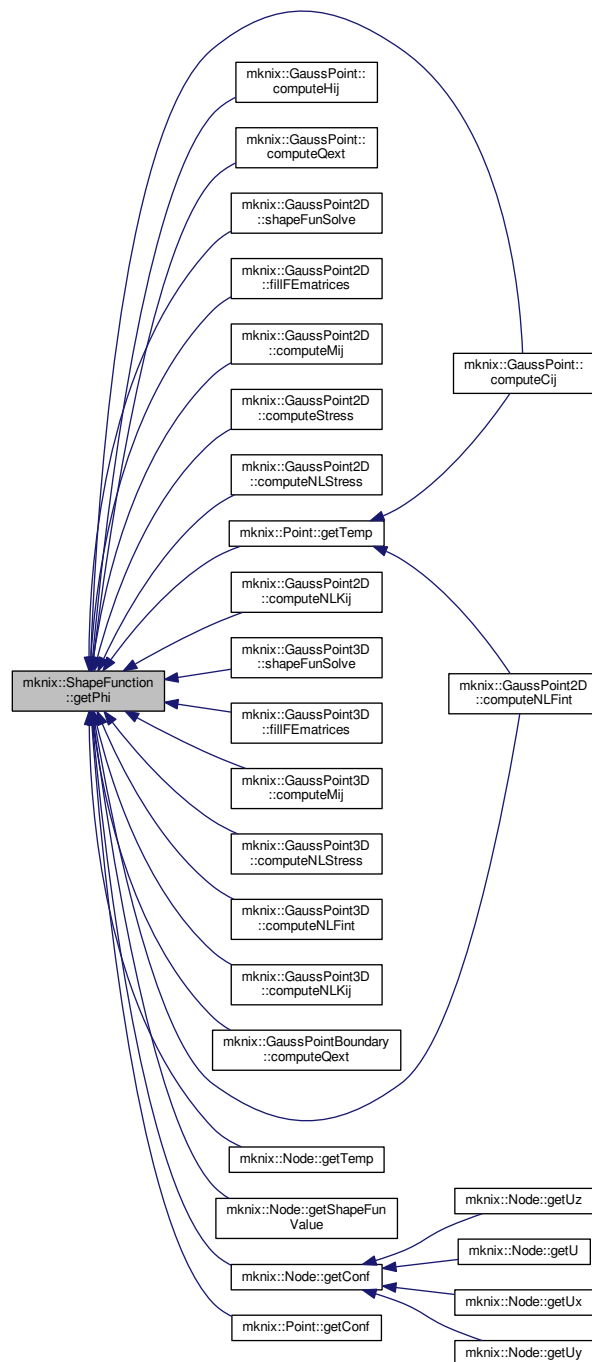
Here is the caller graph for this function:



### 6.55.3.2 virtual double mknix::ShapeFunction::getPhi ( int i, int j ) [inline], [virtual]

Definition at line 61 of file shapefunction.h.

Here is the caller graph for this function:



### 6.55.3.3 void mknix::ShapeFunction::gnuplotOut ( ) [virtual]

Definition at line 60 of file shapefunction.cpp.

Here is the caller graph for this function:



**6.55.3.4** `void mknix::ShapeFunction::outputValues ( )` [virtual]

Definition at line 33 of file `shapefunction.cpp`.

**6.55.3.5** `virtual void mknix::ShapeFunction::setPhi ( double value_in, int i, int j )` [inline], [virtual]

Definition at line 66 of file `shapefunction.h`.

#### 6.55.4 Member Data Documentation

**6.55.4.1** `int mknix::ShapeFunction::dim` [protected]

Definition at line 46 of file `shapefunction.h`.

**6.55.4.2** `Point* mknix::ShapeFunction::gp` [protected]

Definition at line 48 of file `shapefunction.h`.

**6.55.4.3** `Imx::DenseMatrix<double> mknix::ShapeFunction::phi` [protected]

Definition at line 47 of file `shapefunction.h`.

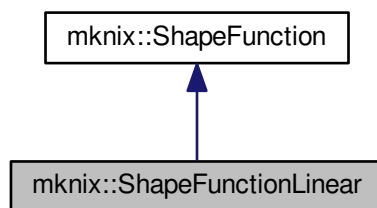
The documentation for this class was generated from the following files:

- [shapefunction.h](#)
- [shapefunction.cpp](#)

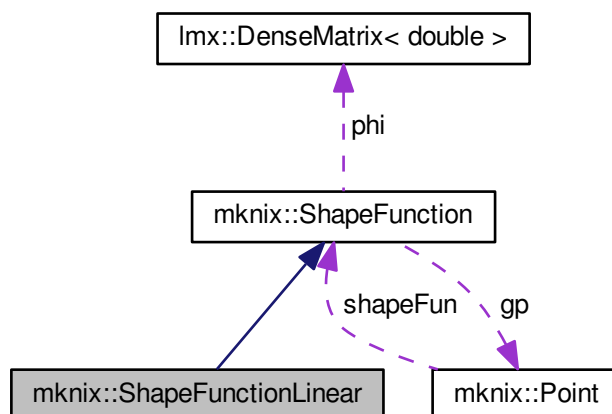
## 6.56 mknix::ShapeFunctionLinear Class Reference

```
#include <shapefunctionlinear.h>
```

Inheritance diagram for `mknix::ShapeFunctionLinear`:



Collaboration diagram for `mknix::ShapeFunctionLinear`:



#### Public Member Functions

- [ShapeFunctionLinear](#) ()
- [ShapeFunctionLinear](#) ([Point](#) \*)
- [~ShapeFunctionLinear](#) ()
- void [calc](#) ()

#### Additional Inherited Members

##### 6.56.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 30 of file `shapefunctionlinear.h`.



### 6.56.2 Constructor & Destructor Documentation

#### 6.56.2.1 mknix::ShapeFunctionLinear::ShapeFunctionLinear ( )

Definition at line 26 of file shapefunctionlinear.cpp.

#### 6.56.2.2 mknix::ShapeFunctionLinear::ShapeFunctionLinear ( Point \* gp\_in )

Definition at line 33 of file shapefunctionlinear.cpp.

#### 6.56.2.3 mknix::ShapeFunctionLinear::~~ShapeFunctionLinear ( )

Definition at line 40 of file shapefunctionlinear.cpp.

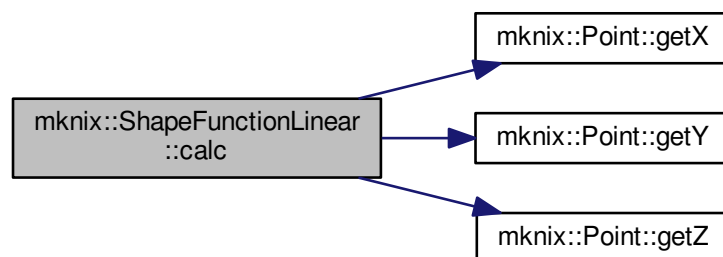
### 6.56.3 Member Function Documentation

#### 6.56.3.1 void mknix::ShapeFunctionLinear::calc ( ) [virtual]

Implements [mnix::ShapeFunction](#).

Definition at line 45 of file shapefunctionlinear.cpp.

Here is the call graph for this function:



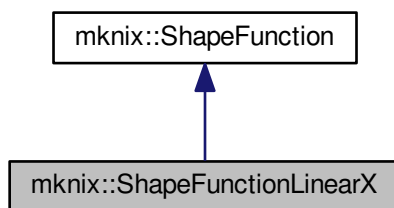
The documentation for this class was generated from the following files:

- [shapefunctionlinear.h](#)
- [shapefunctionlinear.cpp](#)

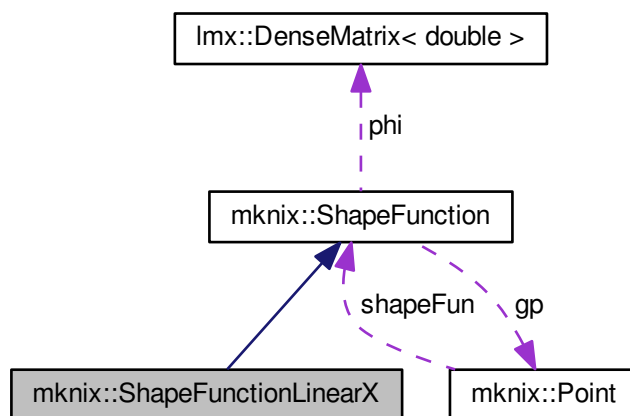
## 6.57 mknix::ShapeFunctionLinearX Class Reference

```
#include <shapefunctionlinear-x.h>
```

Inheritance diagram for `mknix::ShapeFunctionLinearX`:



Collaboration diagram for `mknix::ShapeFunctionLinearX`:



#### Public Member Functions

- [ShapeFunctionLinearX \(\)](#)
- [ShapeFunctionLinearX \(Point \\*\)](#)
- [~ShapeFunctionLinearX \(\)](#)
- void [calc \(\)](#)

#### Additional Inherited Members

##### 6.57.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 30 of file `shapefunctionlinear-x.h`.

### 6.57.2 Constructor & Destructor Documentation

#### 6.57.2.1 mknix::ShapeFunctionLinearX::ShapeFunctionLinearX ( )

Definition at line 26 of file shapefunctionlinear-x.cpp.

#### 6.57.2.2 mknix::ShapeFunctionLinearX::ShapeFunctionLinearX ( Point \* gp\_in )

Definition at line 33 of file shapefunctionlinear-x.cpp.

#### 6.57.2.3 mknix::ShapeFunctionLinearX::~~ShapeFunctionLinearX ( )

Definition at line 40 of file shapefunctionlinear-x.cpp.

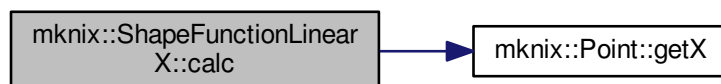
### 6.57.3 Member Function Documentation

#### 6.57.3.1 void mknix::ShapeFunctionLinearX::calc ( ) [virtual]

Implements [mknix::ShapeFunction](#).

Definition at line 45 of file shapefunctionlinear-x.cpp.

Here is the call graph for this function:



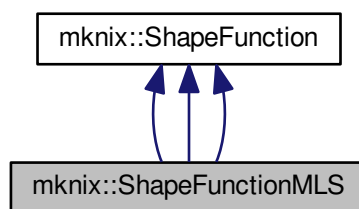
The documentation for this class was generated from the following files:

- [shapefunctionlinear-x.h](#)
- [shapefunctionlinear-x.cpp](#)

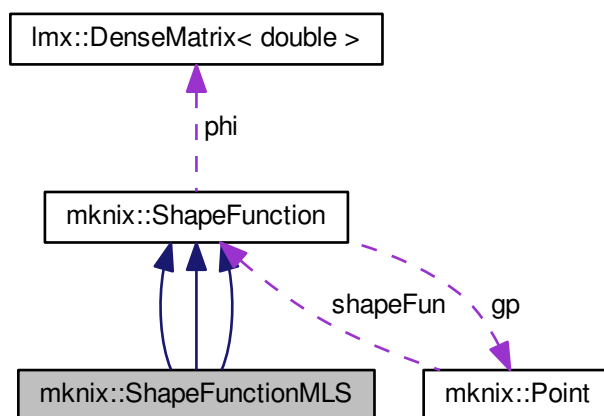
## 6.58 mknix::ShapeFunctionMLS Class Reference

```
#include <shapefunctionMLS.h>
```

Inheritance diagram for `mknix::ShapeFunctionMLS`:



Collaboration diagram for `mknix::ShapeFunctionMLS`:



#### Public Member Functions

- [ShapeFunctionMLS](#) ()
- [ShapeFunctionMLS](#) (int, int, int, double &, double &, [Point](#) \*)
- [~ShapeFunctionMLS](#) ()
- void [calc](#) ()
- [ShapeFunctionMLS](#) ()
- [ShapeFunctionMLS](#) (int, int, int, double &, double &, [Point](#) \*)
- [~ShapeFunctionMLS](#) ()
- void [calc](#) ()
- [ShapeFunctionMLS](#) ()
- [ShapeFunctionMLS](#) (int, int, int, double &, double &, [Point](#) \*)
- [~ShapeFunctionMLS](#) ()
- void [calc](#) ()

## Additional Inherited Members

## 6.58.1 Detailed Description

## Author

Daniel Iglesias

Definition at line 42 of file shapefunctionMLS.h.

## 6.58.2 Constructor &amp; Destructor Documentation

## 6.58.2.1 mknix::ShapeFunctionMLS::ShapeFunctionMLS ( )

Container of B\_I

Definition at line 10 of file shapefunctionMLS.cpp.

6.58.2.2 mknix::ShapeFunctionMLS::ShapeFunctionMLS ( int *nn\_in*, int *mm\_in*, int *weightType\_in*, double & *alpha\_c\_in*, double & *d\_c\_in*, Point \* *gp\_in* )

Definition at line 15 of file shapefunctionMLS.cpp.

## 6.58.2.3 mknix::ShapeFunctionMLS::~~ShapeFunctionMLS ( )

Definition at line 69 of file shapefunctionMLS.cpp.

## 6.58.2.4 mknix::ShapeFunctionMLS::ShapeFunctionMLS ( )

Container of B\_I

## 6.58.2.5 mknix::ShapeFunctionMLS::ShapeFunctionMLS ( int , int , int , double &amp; , double &amp; , Point \* )

## 6.58.2.6 mknix::ShapeFunctionMLS::~~ShapeFunctionMLS ( )

## 6.58.2.7 mknix::ShapeFunctionMLS::ShapeFunctionMLS ( )

Container of B\_I

## 6.58.2.8 mknix::ShapeFunctionMLS::ShapeFunctionMLS ( int , int , int , double &amp; , double &amp; , Point \* )

## 6.58.2.9 mknix::ShapeFunctionMLS::~~ShapeFunctionMLS ( )

## 6.58.3 Member Function Documentation

## 6.58.3.1 void mknix::ShapeFunctionMLS::calc ( ) [virtual]

Implements [mknix::ShapeFunction](#).

Definition at line 73 of file shapefunctionMLS.cpp.

## 6.58.3.2 void mknix::ShapeFunctionMLS::calc ( ) [virtual]

Implements [mknix::ShapeFunction](#).

## 6.58.3.3 void mknix::ShapeFunctionMLS::calc ( ) [virtual]

Implements [mknix::ShapeFunction](#).

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

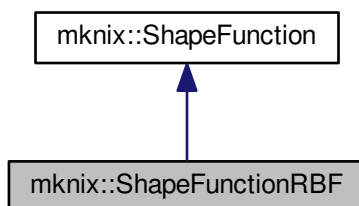
- [shapefunctionMLS.h](#)

- [shapefunctionMLS2D.h](#)
- [shapefunctionMLS3D.h](#)
- [shapefunctionMLS.cpp](#)
- [shapefunctionMLS2D.cpp](#)
- [shapefunctionMLS3D.cpp](#)

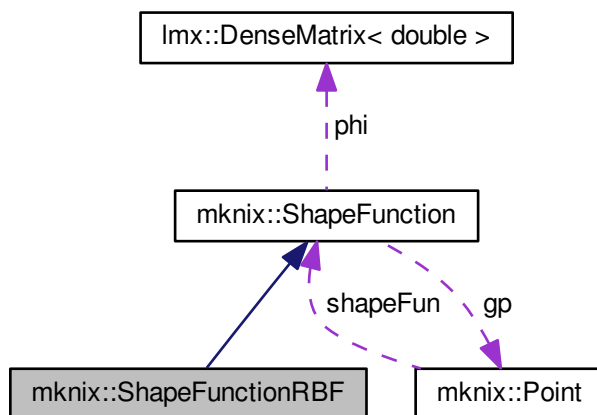
## 6.59 mknix::ShapeFunctionRBF Class Reference

```
#include <shapefunctionRBF.h>
```

Inheritance diagram for mknix::ShapeFunctionRBF:



Collaboration diagram for mknix::ShapeFunctionRBF:



### Public Member Functions

- [ShapeFunctionRBF](#) ()
- [ShapeFunctionRBF](#) (int, int, int, double &, double &, double &, [Point](#) \*)
- [~ShapeFunctionRBF](#) ()
- void [calc](#) ()

## Additional Inherited Members

## 6.59.1 Detailed Description

## Author

Daniel Iglesias

Definition at line 42 of file shapefunctionRBF.h.

## 6.59.2 Constructor &amp; Destructor Documentation

## 6.59.2.1 mknix::ShapeFunctionRBF::ShapeFunctionRBF ( )

Definition at line 10 of file shapefunctionRBF.cpp.

6.59.2.2 mknix::ShapeFunctionRBF::ShapeFunctionRBF ( int *nn\_in*, int *mm\_in*, int *rbfType\_in*, double & *alpha\_c\_in*, double & *d\_c\_in*, double & *q\_in*, Point \* *gp\_in* )

Definition at line 15 of file shapefunctionRBF.cpp.

## 6.59.2.3 mknix::ShapeFunctionRBF::~ShapeFunctionRBF ( )

Definition at line 53 of file shapefunctionRBF.cpp.

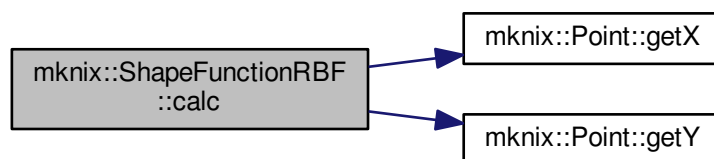
## 6.59.3 Member Function Documentation

## 6.59.3.1 void mknix::ShapeFunctionRBF::calc ( ) [virtual]

Implements [mknix::ShapeFunction](#).

Definition at line 57 of file shapefunctionRBF.cpp.

Here is the call graph for this function:



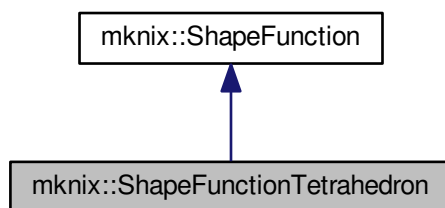
The documentation for this class was generated from the following files:

- [shapefunctionRBF.h](#)
- [shapefunctionRBF.cpp](#)

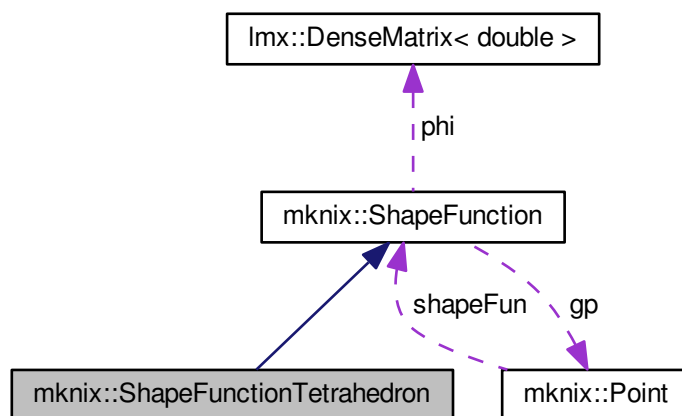
## 6.60 mknix::ShapeFunctionTetrahedron Class Reference

```
#include <shapefunctiontetrahedron.h>
```

Inheritance diagram for `mknix::ShapeFunctionTetrahedron`:



Collaboration diagram for `mknix::ShapeFunctionTetrahedron`:



#### Public Member Functions

- [ShapeFunctionTetrahedron](#) ()
- [ShapeFunctionTetrahedron](#) ([Point](#) \*)
- [~ShapeFunctionTetrahedron](#) ()
- void [calc](#) ()
- void [compute\\_abcd](#) (int, int, int)

#### Additional Inherited Members

##### 6.60.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 30 of file `shapefunctiontetrahedron.h`.



## 6.60.2 Constructor &amp; Destructor Documentation

## 6.60.2.1 mknix::ShapeFunctionTetrahedron::ShapeFunctionTetrahedron ( )

Definition at line 26 of file shapefunctiontetrahedron.cpp.

## 6.60.2.2 mknix::ShapeFunctionTetrahedron::ShapeFunctionTetrahedron ( Point \* gp\_in )

Definition at line 32 of file shapefunctiontetrahedron.cpp.

## 6.60.2.3 mknix::ShapeFunctionTetrahedron::~~ShapeFunctionTetrahedron ( )

Definition at line 41 of file shapefunctiontetrahedron.cpp.

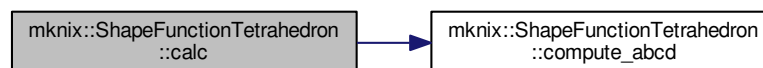
## 6.60.3 Member Function Documentation

## 6.60.3.1 void mknix::ShapeFunctionTetrahedron::calc ( ) [virtual]

Implements [mknix::ShapeFunction](#).

Definition at line 46 of file shapefunctiontetrahedron.cpp.

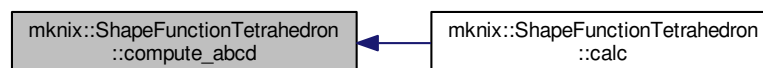
Here is the call graph for this function:



## 6.60.3.2 void mknix::ShapeFunctionTetrahedron::compute\_abcd ( int i1, int i2, int i3 )

Definition at line 84 of file shapefunctiontetrahedron.cpp.

Here is the caller graph for this function:



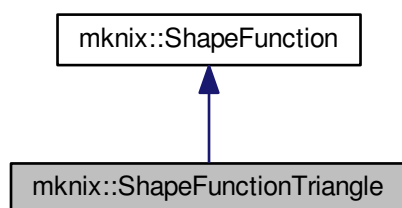
The documentation for this class was generated from the following files:

- [shapefunctiontetrahedron.h](#)
- [shapefunctiontetrahedron.cpp](#)

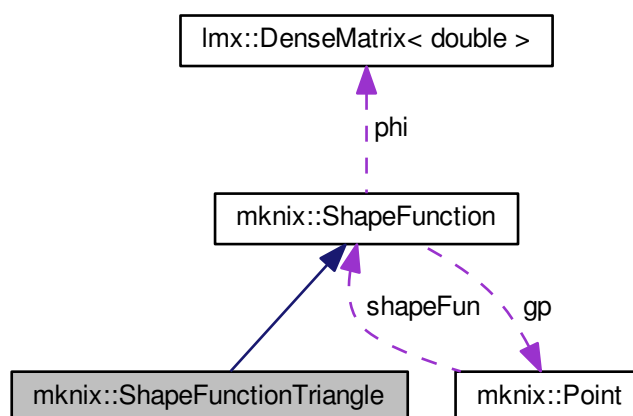
## 6.61 mknix::ShapeFunctionTriangle Class Reference

```
#include <shapefunctiontriangle.h>
```

Inheritance diagram for `mknix::ShapeFunctionTriangle`:



Collaboration diagram for `mknix::ShapeFunctionTriangle`:



#### Public Member Functions

- [ShapeFunctionTriangle \(\)](#)
- [ShapeFunctionTriangle \(Point \\*\)](#)
- [~ShapeFunctionTriangle \(\)](#)
- void [calc \(\)](#)

#### Additional Inherited Members

##### 6.61.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 30 of file `shapefunctiontriangle.h`.

### 6.61.2 Constructor & Destructor Documentation

#### 6.61.2.1 mknix::ShapeFunctionTriangle::ShapeFunctionTriangle ( )

Definition at line 28 of file shapefunctiontriangle.cpp.

#### 6.61.2.2 mknix::ShapeFunctionTriangle::ShapeFunctionTriangle ( Point \* gp\_in )

Definition at line 35 of file shapefunctiontriangle.cpp.

#### 6.61.2.3 mknix::ShapeFunctionTriangle::~~ShapeFunctionTriangle ( )

Definition at line 42 of file shapefunctiontriangle.cpp.

### 6.61.3 Member Function Documentation

#### 6.61.3.1 void mknix::ShapeFunctionTriangle::calc ( ) [virtual]

Implements [mknix::ShapeFunction](#).

Definition at line 47 of file shapefunctiontriangle.cpp.

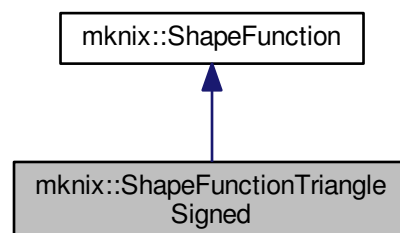
The documentation for this class was generated from the following files:

- [shapefunctiontriangle.h](#)
- [shapefunctiontriangle.cpp](#)

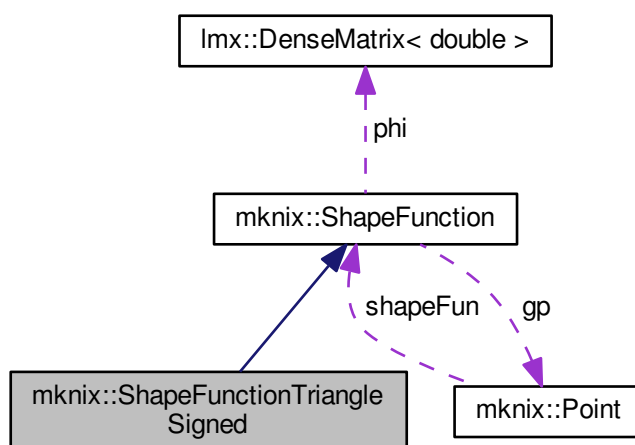
## 6.62 mknix::ShapeFunctionTriangleSigned Class Reference

```
#include <shapefunctiontriangle3D.h>
```

Inheritance diagram for mknix::ShapeFunctionTriangleSigned:



Collaboration diagram for `mnix::ShapeFunctionTriangleSigned`:



#### Public Member Functions

- [ShapeFunctionTriangleSigned \(\)](#)
- [ShapeFunctionTriangleSigned \(Point \\*\)](#)
- [~ShapeFunctionTriangleSigned \(\)](#)
- void [calc \(\)](#)

#### Additional Inherited Members

##### 6.62.1 Detailed Description

###### Author

AUTHORS <MAILS>

Definition at line 30 of file `shapefunctiontriangle3D.h`.

##### 6.62.2 Constructor & Destructor Documentation

###### 6.62.2.1 `mnix::ShapeFunctionTriangleSigned::ShapeFunctionTriangleSigned ( )`

Definition at line 26 of file `shapefunctiontriangle3D.cpp`.

###### 6.62.2.2 `mnix::ShapeFunctionTriangleSigned::ShapeFunctionTriangleSigned ( Point * gp_in )`

Definition at line 33 of file `shapefunctiontriangle3D.cpp`.

###### 6.62.2.3 `mnix::ShapeFunctionTriangleSigned::~~ShapeFunctionTriangleSigned ( )`

Definition at line 40 of file `shapefunctiontriangle3D.cpp`.

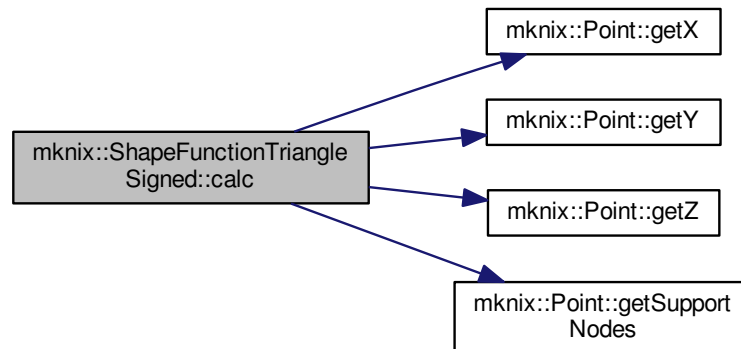
## 6.62.3 Member Function Documentation

## 6.62.3.1 void mknix::ShapeFunctionTriangleSigned::calc ( ) [virtual]

Implements [mknix::ShapeFunction](#).

Definition at line 45 of file `shapefunctiontriangle3D.cpp`.

Here is the call graph for this function:



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

- [shapefunctiontriangle3D.h](#)
- [shapefunctiontriangle3D.cpp](#)

## 6.63 mknix::Simulation Class Reference

```
#include <simulation.h>
```

## Public Member Functions

- [Simulation](#) ( )
- [~Simulation](#) ( )
- [Simulation](#) (const [Simulation](#) &)=delete
- [Simulation](#) & [operator=](#) (const [Simulation](#) &)=delete
- void [inputFromFile](#) (const std::string &fileIn)
- std::vector< double > [getInterfaceNodesCoords](#) ( )
- void [setOutputFilesDetail](#) (int level\_in)
- void [init](#) ( )
- void [setInitialTemperatures](#) (double)
- void [solveStep](#) ( )
- void [solveStep](#) (double \*, double \*o\_output=0)
- void [endSimulation](#) ( )
- void [run](#) ( )
- void [runThermalAnalysis](#) ([Analysis](#) \*)
- void [runMechanicalAnalysis](#) ([Analysis](#) \*)
- void [writeSystem](#) ( )

- void [staticThermalResidue](#) (Imx::Vector< [data\\_type](#) > &residue, Imx::Vector< [data\\_type](#) > &q)
- void [staticThermalTangent](#) (Imx::Matrix< [data\\_type](#) > &tangent\_in, Imx::Vector< [data\\_type](#) > &q)
- bool [staticThermalConvergence](#) (Imx::Vector< [data\\_type](#) > &res, Imx::Vector< [data\\_type](#) > &q)
- void [explicitThermalEvaluation](#) (const Imx::Vector< [data\\_type](#) > &q, Imx::Vector< [data\\_type](#) > &qdot, double time)
- void [dynamicThermalEvaluation](#) (const Imx::Vector< [data\\_type](#) > &q, Imx::Vector< [data\\_type](#) > &qdot, double time)
- void [dynamicThermalResidue](#) (Imx::Vector< [data\\_type](#) > &residue, const Imx::Vector< [data\\_type](#) > &q, const Imx::Vector< [data\\_type](#) > &qdot, double time)
- void [dynamicThermalTangent](#) (Imx::Matrix< [data\\_type](#) > &tangent\_in, const Imx::Vector< [data\\_type](#) > &q, double partial\_qdot, double time)
- bool [dynamicThermalConvergence](#) (const Imx::Vector< [data\\_type](#) > &q, const Imx::Vector< [data\\_type](#) > &qdot, double time)
- bool [dynamicThermalConvergenceInThermomechanical](#) (const Imx::Vector< [data\\_type](#) > &q, const Imx::Vector< [data\\_type](#) > &qdot, double time)
- void [explicitAcceleration](#) (const Imx::Vector< [data\\_type](#) > &q, const Imx::Vector< [data\\_type](#) > &qdot, Imx::Vector< [data\\_type](#) > &qddot, double time)
- void [dynamicAcceleration](#) (const Imx::Vector< [data\\_type](#) > &q, const Imx::Vector< [data\\_type](#) > &qdot, Imx::Vector< [data\\_type](#) > &qddot, double time)
- void [dynamicResidue](#) (Imx::Vector< [data\\_type](#) > &residue, const Imx::Vector< [data\\_type](#) > &q, const Imx::Vector< [data\\_type](#) > &qdot, const Imx::Vector< [data\\_type](#) > &qddot, double time)
- void [dynamicTangent](#) (Imx::Matrix< [data\\_type](#) > &tangent\_in, const Imx::Vector< [data\\_type](#) > &q, const Imx::Vector< [data\\_type](#) > &qdot, double partial\_qdot, double partial\_qddot, double time)
- bool [dynamicConvergence](#) (const Imx::Vector< [data\\_type](#) > &q, const Imx::Vector< [data\\_type](#) > &qdot, const Imx::Vector< [data\\_type](#) > &qddot, double time)
- void [staticResidue](#) (Imx::Vector< [data\\_type](#) > &residue, Imx::Vector< [data\\_type](#) > &q)
- void [staticTangent](#) (Imx::Matrix< [data\\_type](#) > &tangent\_in, Imx::Vector< [data\\_type](#) > &q)
- bool [staticConvergence](#) (Imx::Vector< [data\\_type](#) > &res, Imx::Vector< [data\\_type](#) > &q)
- void [stepTriggered](#) ()
- void [writeConfStep](#) ()
- Imx::DenseMatrix< [data\\_type](#) > & [getSparsePattern](#) ()

#### Static Public Member Functions

- static double [getGravity](#) (int component)
- static double [getAlpha](#) ()
- static double [getTime](#) ()
- static int [getDim](#) ()
- static std::string [getConstraintMethod](#) ()
- static std::string [getSmoothingType](#) ()

#### Friends

- class [Reader](#)
- class [ReaderConstraints](#)
- class [ReaderFlex](#)
- class [ReaderRigid](#)
- class [Contact](#)
- class [SystemChain](#)

#### 6.63.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 46 of file simulation.h.

## 6.63.2 Constructor &amp; Destructor Documentation

## 6.63.2.1 mknix::Simulation::Simulation ( )

Definition at line 44 of file simulation.cpp.

## 6.63.2.2 mknix::Simulation::~~Simulation ( )

Definition at line 64 of file simulation.cpp.

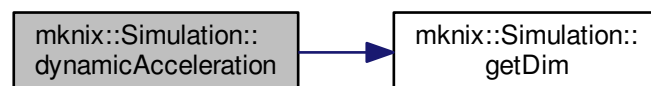
## 6.63.2.3 mknix::Simulation::Simulation ( const Simulation &amp; ) [delete]

## 6.63.3 Member Function Documentation

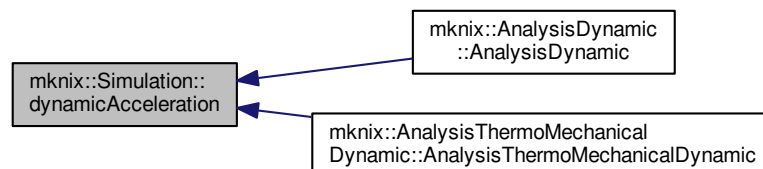
## 6.63.3.1 void mknix::Simulation::dynamicAcceleration ( const Imx::Vector&lt; data\_type &gt; &amp; q, const Imx::Vector&lt; data\_type &gt; &amp; qdot, Imx::Vector&lt; data\_type &gt; &amp; qddot, double time )

Definition at line 736 of file simulation.cpp.

Here is the call graph for this function:



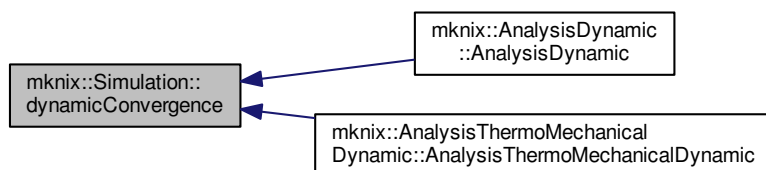
Here is the caller graph for this function:



## 6.63.3.2 bool mknix::Simulation::dynamicConvergence ( const Imx::Vector&lt; data\_type &gt; &amp; q, const Imx::Vector&lt; data\_type &gt; &amp; qdot, const Imx::Vector&lt; data\_type &gt; &amp; qddot, double time )

Definition at line 814 of file simulation.cpp.

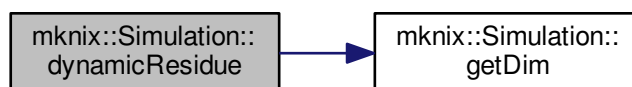
Here is the caller graph for this function:



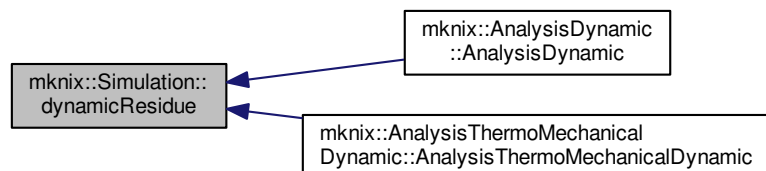
6.63.3.3 `void mknix::Simulation::dynamicResidue ( Imx::Vector< data_type > & residue, const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot, const Imx::Vector< data_type > & qddot, double time )`

Definition at line 764 of file simulation.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

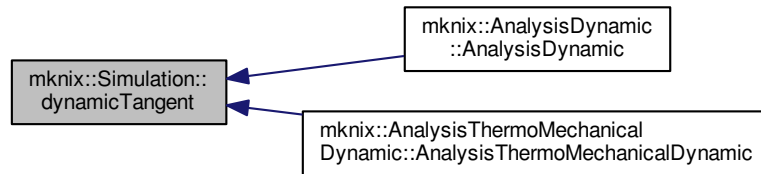


6.63.3.4 `void mknix::Simulation::dynamicTangent ( Imx::Matrix< data_type > & tangent_in, const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot, double partial_qdot, double partial_qddot, double time )`

Definition at line 800 of file simulation.cpp.



Here is the caller graph for this function:



**6.63.3.5** `bool mknix::Simulation::dynamicThermalConvergence ( const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot, double time )`

Definition at line 649 of file simulation.cpp.

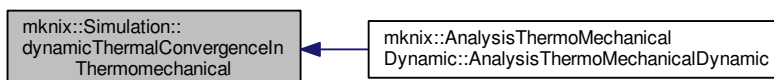
Here is the caller graph for this function:



**6.63.3.6** `bool mknix::Simulation::dynamicThermalConvergenceInThermomechanical ( const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot, double time )`

Definition at line 685 of file simulation.cpp.

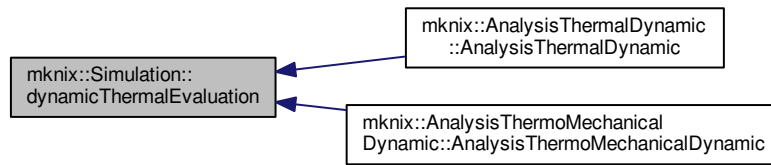
Here is the caller graph for this function:



**6.63.3.7** `void mknix::Simulation::dynamicThermalEvaluation ( const Imx::Vector< data_type > & q, Imx::Vector< data_type > & qdot, double time )`

Definition at line 561 of file simulation.cpp.

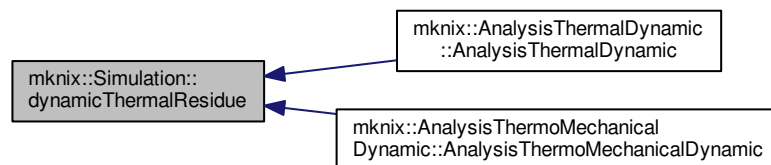
Here is the caller graph for this function:



**6.63.3.8** `void mknix::Simulation::dynamicThermalResidue ( Imx::Vector< data_type > & residue, const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot, double time )`

Definition at line 593 of file simulation.cpp.

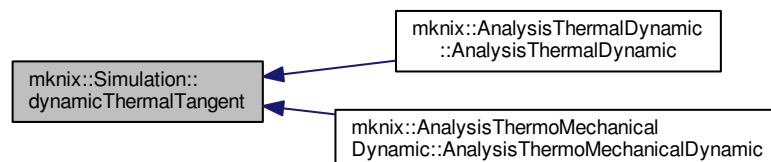
Here is the caller graph for this function:



**6.63.3.9** `void mknix::Simulation::dynamicThermalTangent ( Imx::Matrix< data_type > & tangent_in, const Imx::Vector< data_type > & q, double partial_qdot, double time )`

Definition at line 636 of file simulation.cpp.

Here is the caller graph for this function:



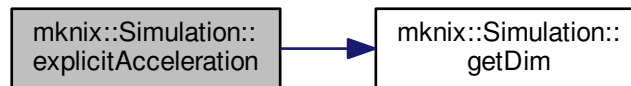
**6.63.3.10** `void mknix::Simulation::endSimulation ( )`

Definition at line 227 of file simulation.cpp.

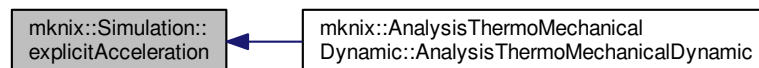
6.63.3.11 `void mknix::Simulation::explicitAcceleration ( const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot, Imx::Vector< data_type > & qddot, double time )`

Definition at line 704 of file simulation.cpp.

Here is the call graph for this function:



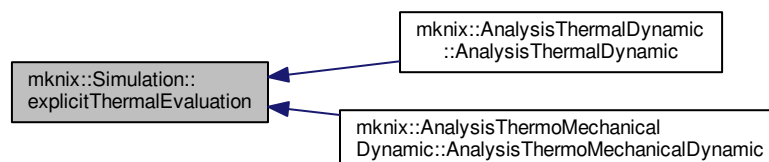
Here is the caller graph for this function:



6.63.3.12 `void mknix::Simulation::explicitThermalEvaluation ( const Imx::Vector< data_type > & qt, Imx::Vector< data_type > & qtdot, double time )`

Definition at line 531 of file simulation.cpp.

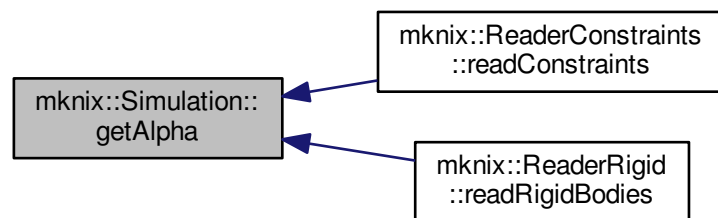
Here is the caller graph for this function:



6.63.3.13 `static double mknix::Simulation::getAlpha ( ) [inline], [static]`

Definition at line 200 of file simulation.h.

Here is the caller graph for this function:



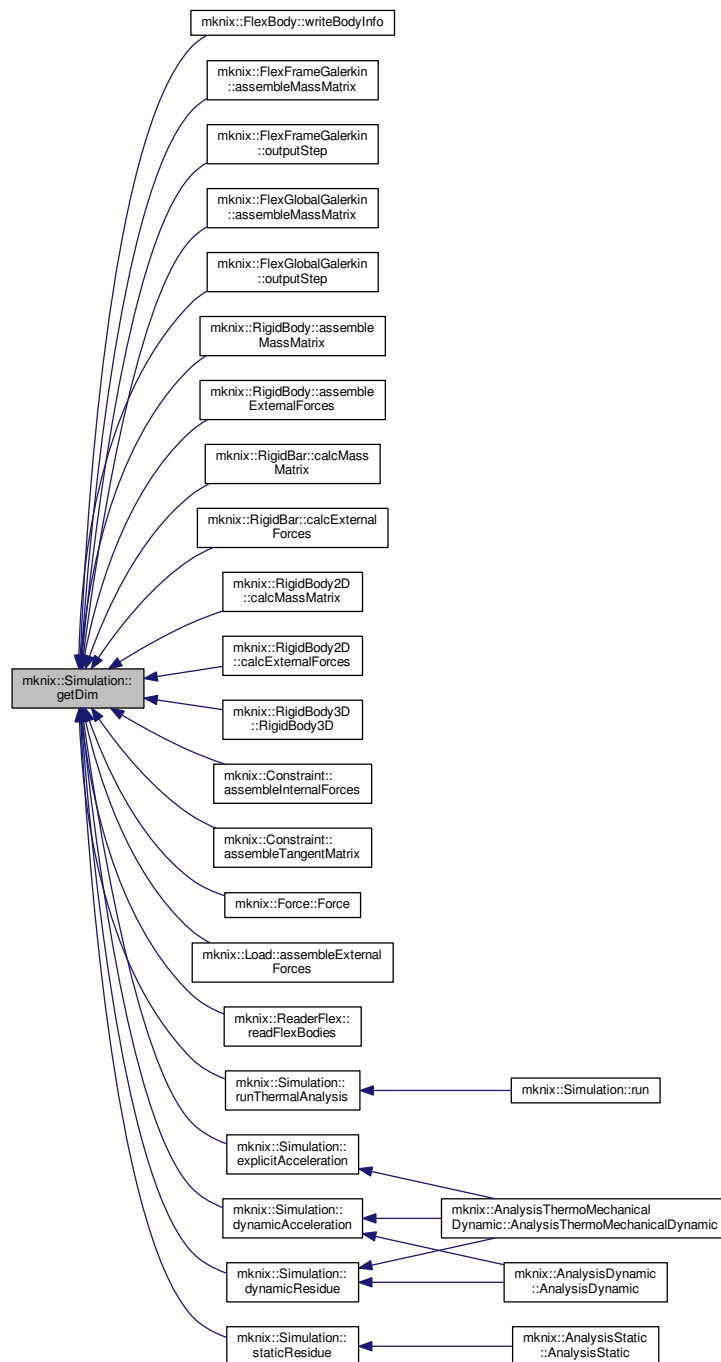
**6.63.3.14** `static std::string mknix::Simulation::getConstraintMethod ( ) [inline],[static]`

Definition at line 215 of file simulation.h.

**6.63.3.15** `static int mknix::Simulation::getDim ( ) [inline],[static]`

Definition at line 210 of file simulation.h.

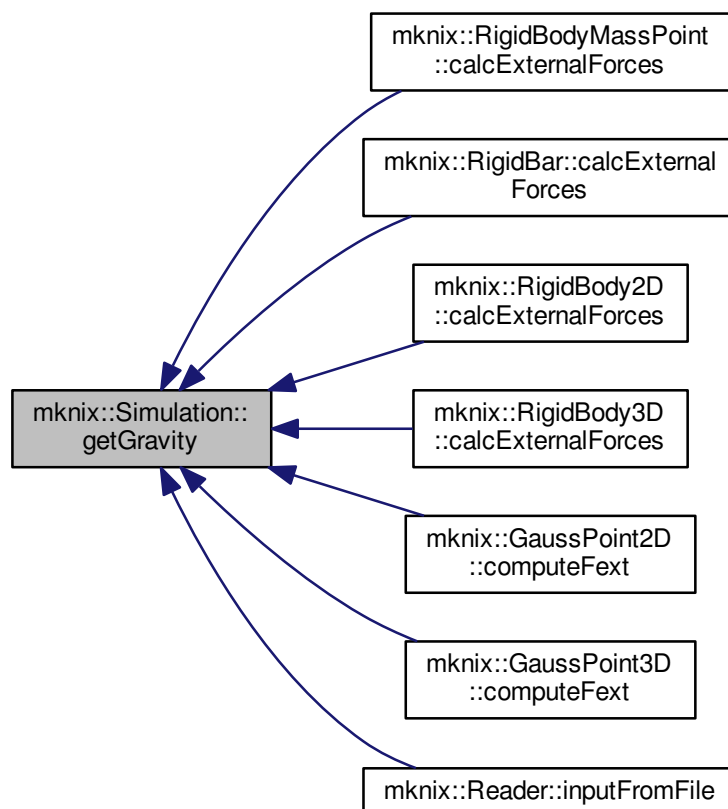
Here is the caller graph for this function:



6.63.3.16 `static double mknix::Simulation::getGravity ( int component )` `[inline],[static]`

Definition at line 195 of file `simulation.h`.

Here is the caller graph for this function:



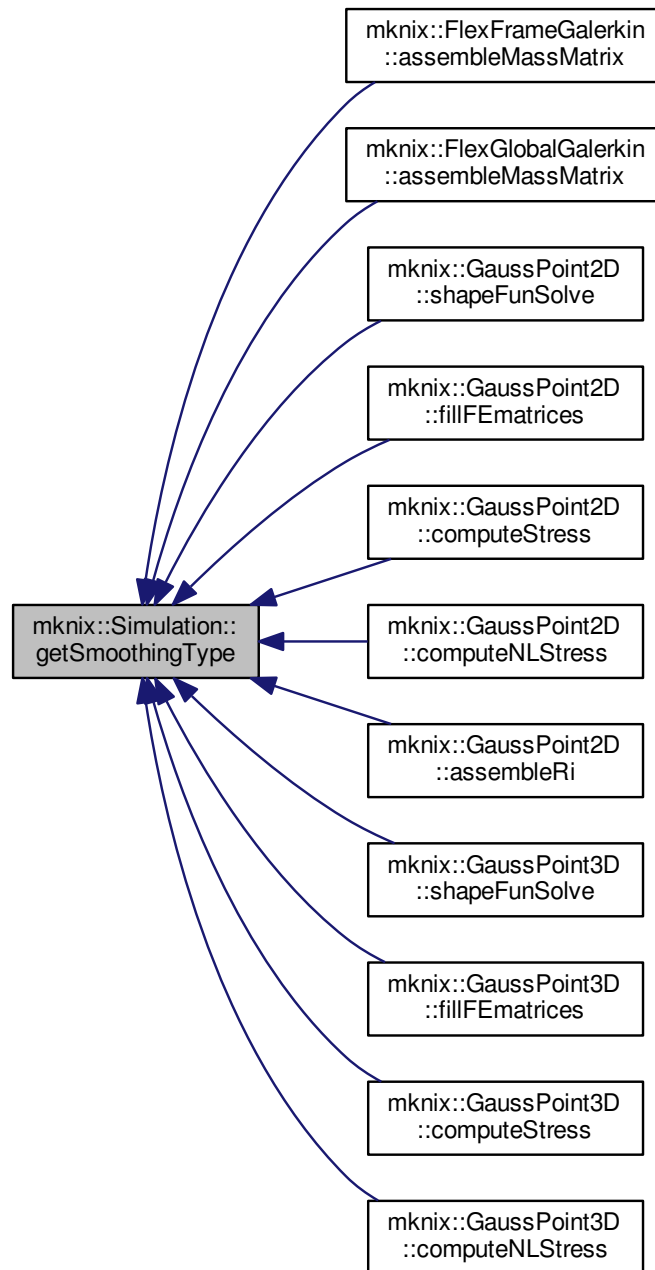
**6.63.3.17** `std::vector< double > mknix::Simulation::getInterfaceNodesCoords ( )`

Definition at line 81 of file `simulation.cpp`.

**6.63.3.18** `static std::string mknix::Simulation::getSmoothingType ( ) [inline], [static]`

Definition at line 220 of file `simulation.h`.

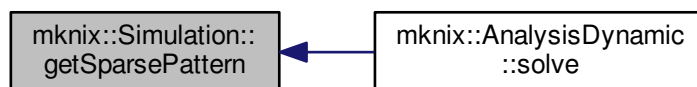
Here is the caller graph for this function:



6.63.3.19 `Imx::DenseMatrix<data_type>& mknix::Simulation::getSparsePattern ( )` [inline]

Definition at line 190 of file `simulation.h`.

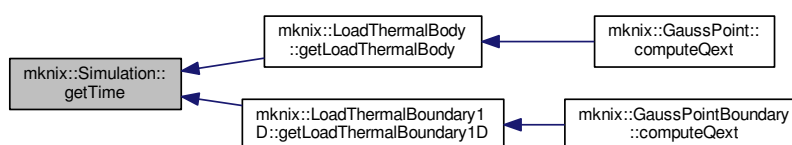
Here is the caller graph for this function:



**6.63.3.20** `static double mknix::Simulation::getTime ( ) [inline],[static]`

Definition at line 205 of file `simulation.h`.

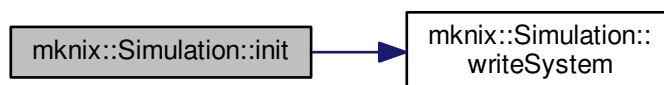
Here is the caller graph for this function:



**6.63.3.21** `void mknix::Simulation::init ( )`

Definition at line 97 of file `simulation.cpp`.

Here is the call graph for this function:



**6.63.3.22** `void mknix::Simulation::inputFromFile ( const std::string & fileName )`

Definition at line 72 of file `simulation.cpp`.

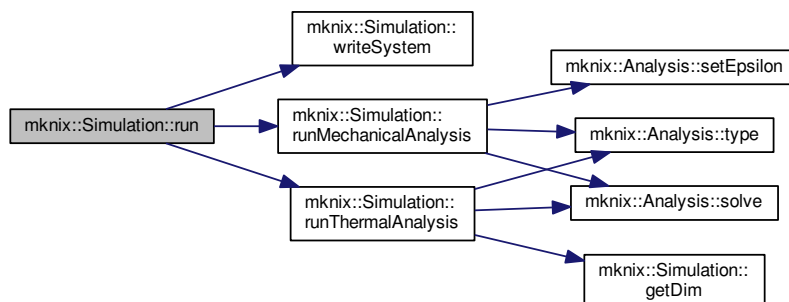
**6.63.3.23** `Simulation& mknix::Simulation::operator=( const Simulation & ) [delete]`

**6.63.3.24** `void mknix::Simulation::run ( )`

Definition at line 269 of file `simulation.cpp`.



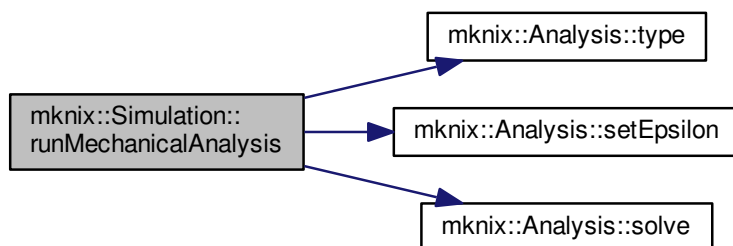
Here is the call graph for this function:



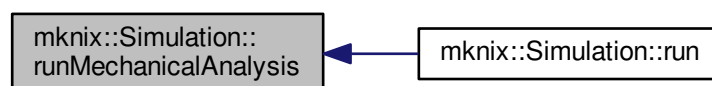
#### 6.63.3.25 void mknix::Simulation::runMechanicalAnalysis ( Analysis \* theAnalysis\_in )

Definition at line 351 of file simulation.cpp.

Here is the call graph for this function:



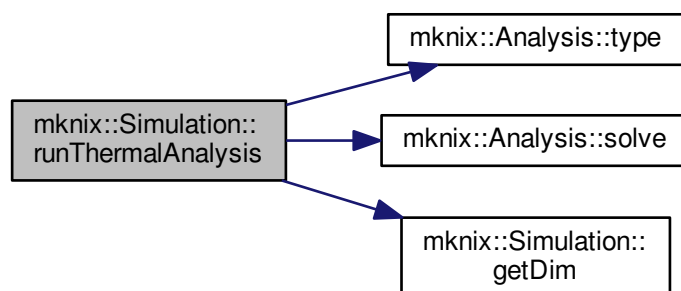
Here is the caller graph for this function:



#### 6.63.3.26 void mknix::Simulation::runThermalAnalysis ( Analysis \* theAnalysis\_in )

Definition at line 303 of file simulation.cpp.

Here is the call graph for this function:



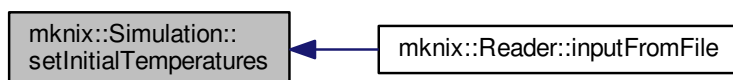
Here is the caller graph for this function:



**6.63.3.27** `void mknix::Simulation::setInitialTemperatures ( double temp_in )`

Definition at line 91 of file `simulation.cpp`.

Here is the caller graph for this function:



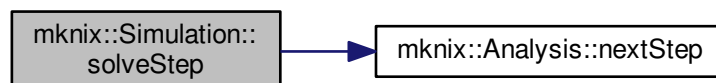
**6.63.3.28** `void mknix::Simulation::setOutputFilesDetail ( int level_in ) [inline]`

Definition at line 76 of file `simulation.h`.

**6.63.3.29** `void mknix::Simulation::solveStep ( )`

Definition at line 213 of file `simulation.cpp`.

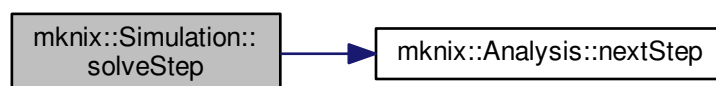
Here is the call graph for this function:



6.63.3.30 `void mknix::Simulation::solveStep ( double * signal, double * o_output = 0 )`

Definition at line 218 of file `simulation.cpp`.

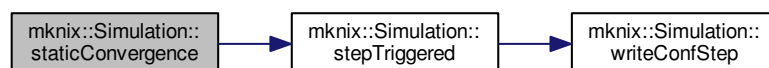
Here is the call graph for this function:



6.63.3.31 `bool mknix::Simulation::staticConvergence ( Imx::Vector< data_type > & res, Imx::Vector< data_type > & q )`

Definition at line 892 of file `simulation.cpp`.

Here is the call graph for this function:



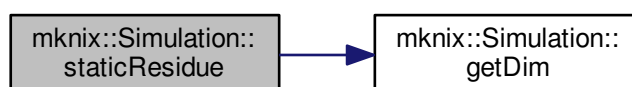
Here is the caller graph for this function:



6.63.3.32 `void mknix::Simulation::staticResidue ( Imx::Vector< data_type > & residue, Imx::Vector< data_type > & q )`

Definition at line 856 of file simulation.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.63.3.33 `void mknix::Simulation::staticTangent ( Imx::Matrix< data_type > & tangent_in, Imx::Vector< data_type > & q )`

Definition at line 882 of file simulation.cpp.

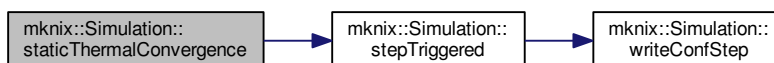
Here is the caller graph for this function:



6.63.3.34 `bool mknix::Simulation::staticThermalConvergence ( Imx::Vector< data_type > & res, Imx::Vector< data_type > & q )`

Definition at line 509 of file simulation.cpp.

Here is the call graph for this function:



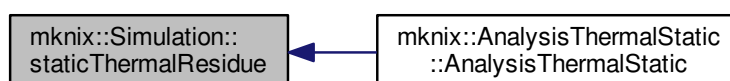
Here is the caller graph for this function:



6.63.3.35 `void mknix::Simulation::staticThermalResidue ( Imx::Vector< data_type > & residue, Imx::Vector< data_type > & q )`

Definition at line 469 of file simulation.cpp.

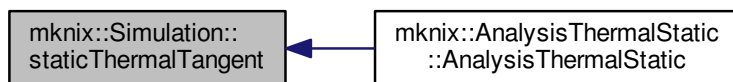
Here is the caller graph for this function:



6.63.3.36 `void mknix::Simulation::staticThermalTangent ( Imx::Matrix< data_type > & tangent_in, Imx::Vector< data_type > & q )`

Definition at line 497 of file simulation.cpp.

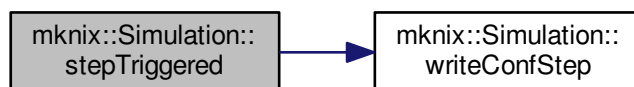
Here is the caller graph for this function:



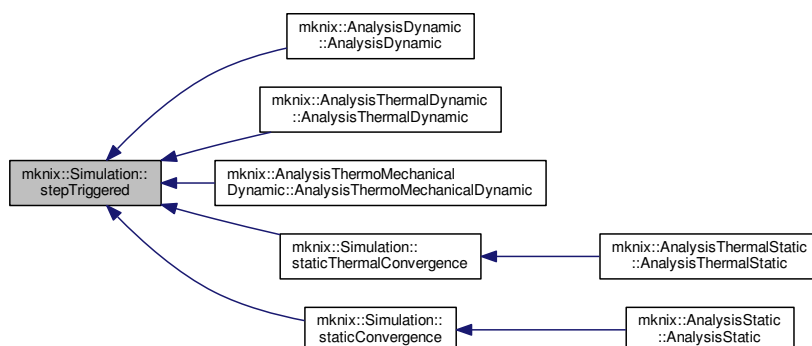
6.63.3.37 `void mknix::Simulation::stepTriggered ( )`

Definition at line 914 of file simulation.cpp.

Here is the call graph for this function:



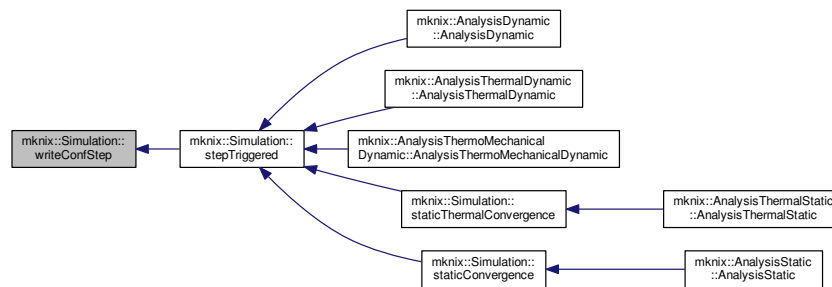
Here is the caller graph for this function:



6.63.3.38 `void mknix::Simulation::writeConfStep ( )`

Definition at line 940 of file simulation.cpp.

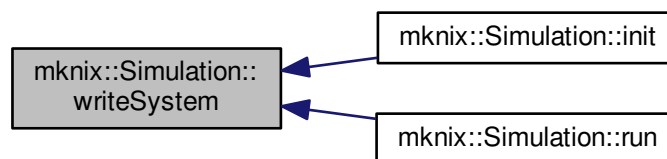
Here is the caller graph for this function:



#### 6.63.3.39 void mknix::Simulation::writeSystem ( )

Definition at line 421 of file simulation.cpp.

Here is the caller graph for this function:



### 6.63.4 Friends And Related Function Documentation

#### 6.63.4.1 friend class Contact [friend]

Definition at line 57 of file simulation.h.

#### 6.63.4.2 friend class Reader [friend]

Definition at line 49 of file simulation.h.

#### 6.63.4.3 friend class ReaderConstraints [friend]

Definition at line 51 of file simulation.h.

#### 6.63.4.4 friend class ReaderFlex [friend]

Definition at line 53 of file simulation.h.

#### 6.63.4.5 friend class ReaderRigid [friend]

Definition at line 55 of file simulation.h.

#### 6.63.4.6 friend class **SystemChain** [friend]

Definition at line 59 of file simulation.h.

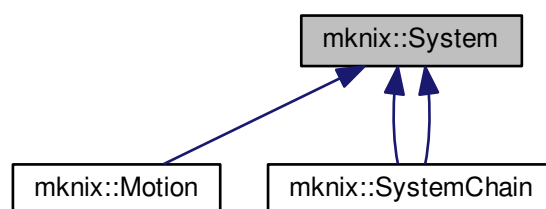
The documentation for this class was generated from the following files:

- [simulation.h](#)
- [simulation.cpp](#)

### 6.64 mknix::System Class Reference

```
#include <system.h>
```

Inheritance diagram for mknix::System:



#### Public Member Functions

- [System](#) ()
- [System](#) (const std::string &title)
- virtual [~System](#) ()
- std::string [getTitle](#) ()
- virtual [Node](#) \* [getNode](#) (int index)
- void [getThermalNodes](#) (std::vector< double > &)
- void [getOutputSignalThermal](#) (double \*)
- void [updateThermalLoads](#) (double \*)
- virtual void [update](#) (double)
- void [initFlexBodies](#) ()
- void [writeRigidBodies](#) (std::ofstream \*)
- void [writeFlexBodies](#) (std::ofstream \*)
- void [writeJoints](#) (std::ofstream \*)
- void [calcCapacityMatrix](#) ()
- void [calcConductivityMatrix](#) ()
- void [calcInternalHeat](#) ()
- void [calcExternalHeat](#) ()
- void [calcThermalTangentMatrix](#) ()
- void [assembleCapacityMatrix](#) (Imx::Matrix< data\_type > &)
- void [assembleConductivityMatrix](#) (Imx::Matrix< data\_type > &)
- void [assembleExternalHeat](#) (Imx::Vector< data\_type > &)
- void [assembleInternalHeat](#) (Imx::Vector< data\_type > &)
- void [assembleThermalTangentMatrix](#) (Imx::Matrix< data\_type > &)
- void [calcMassMatrix](#) ()



- void [calcInternalForces](#) ()
- void [calcExternalForces](#) ()
- void [calcTangentMatrix](#) ()
- void [assembleMassMatrix](#) (Imx::Matrix< data\_type > &)
- void [assembleInternalForces](#) (Imx::Vector< data\_type > &)
- void [assembleExternalForces](#) (Imx::Vector< data\_type > &)
- void [assembleTangentMatrix](#) (Imx::Matrix< data\_type > &)
- void [assembleConstraintForces](#) (Imx::Vector< data\_type > &)
- void [setMechanical](#) ()
- void [outputStep](#) (const Imx::Vector< data\_type > &, const Imx::Vector< data\_type > &)
- void [outputStep](#) (const Imx::Vector< data\_type > &)
- void [outputToFile](#) (std::ofstream \*)
- bool [checkAugmented](#) ()
- void [clearAugmented](#) ()
- void [writeBoundaryNodes](#) (std::vector< Point \* > &)
- void [writeBoundaryConnectivity](#) (std::vector< std::vector< Point \* > > &)

#### Public Attributes

- bool [outputMaxInterfaceTemp](#)

#### Protected Attributes

- std::string [title](#)
- std::vector< [Node](#) \* > [groundNodes](#)
- std::map< std::string, [System](#) \* > [subSystems](#)
- std::map< std::string, [RigidBody](#) \* > [rigidBodies](#)
- std::map< std::string, [FlexBody](#) \* > [flexBodies](#)
- std::map< std::string, [Body](#) \* > [thermalBodies](#)
- std::map< std::string, [Constraint](#) \* > [constraints](#)
- std::map< std::string, [ConstraintThermal](#) \* > [constraintsThermal](#)
- std::vector< [Load](#) \* > [loads](#)
- std::vector< [LoadThermal](#) \* > [loadsThermal](#)
- std::vector< [Node](#) \* > [outputSignalThermal](#)
- std::vector< [Motion](#) \* > [motions](#)

#### Friends

- class [Reader](#)
- class [ReaderFlex](#)
- class [ReaderRigid](#)
- class [ReaderConstraints](#)
- class [Contact](#)

#### 6.64.1 Detailed Description

##### Author

AUTHORS <MAILS>

Definition at line 44 of file system.h.

## 6.64.2 Constructor & Destructor Documentation

### 6.64.2.1 `mknix::System::System ( )`

Definition at line 33 of file `system.cpp`.

### 6.64.2.2 `mknix::System::System ( const std::string & title )`

Definition at line 39 of file `system.cpp`.

### 6.64.2.3 `mknix::System::~~System ( )` `[virtual]`

Definition at line 46 of file `system.cpp`.

## 6.64.3 Member Function Documentation

### 6.64.3.1 `void mknix::System::assembleCapacityMatrix ( Imx::Matrix< data_type > & globalCapacity_in )`

Definition at line 232 of file `system.cpp`.

### 6.64.3.2 `void mknix::System::assembleConductivityMatrix ( Imx::Matrix< data_type > & globalConductivity_in )`

Definition at line 244 of file `system.cpp`.

### 6.64.3.3 `void mknix::System::assembleConstraintForces ( Imx::Vector< data_type > & internalForces_in )`

Definition at line 440 of file `system.cpp`.

### 6.64.3.4 `void mknix::System::assembleExternalForces ( Imx::Vector< data_type > & externalForces_in )`

Definition at line 401 of file `system.cpp`.

### 6.64.3.5 `void mknix::System::assembleExternalHeat ( Imx::Vector< data_type > & externalHeat_in )`

Definition at line 264 of file `system.cpp`.

### 6.64.3.6 `void mknix::System::assembleInternalForces ( Imx::Vector< data_type > & internalForces_in )`

Definition at line 385 of file `system.cpp`.

### 6.64.3.7 `void mknix::System::assembleInternalHeat ( Imx::Vector< data_type > & internalHeat_in )`

Definition at line 281 of file `system.cpp`.

### 6.64.3.8 `void mknix::System::assembleMassMatrix ( Imx::Matrix< data_type > & globalMass_in )`

Definition at line 369 of file `system.cpp`.

### 6.64.3.9 `void mknix::System::assembleTangentMatrix ( Imx::Matrix< data_type > & globalTangent_in )`

Definition at line 424 of file `system.cpp`.

### 6.64.3.10 `void mknix::System::assembleThermalTangentMatrix ( Imx::Matrix< data_type > & globalTangent_in )`

Definition at line 293 of file `system.cpp`.

### 6.64.3.11 `void mknix::System::calcCapacityMatrix ( )`

Definition at line 167 of file `system.cpp`.

**6.64.3.12 void mknix::System::calcConductivityMatrix ( )**

Definition at line 178 of file system.cpp.

**6.64.3.13 void mknix::System::calcExternalForces ( )**

Definition at line 336 of file system.cpp.

**6.64.3.14 void mknix::System::calcExternalHeat ( )**

Definition at line 197 of file system.cpp.

**6.64.3.15 void mknix::System::calcInternalForces ( )**

Definition at line 320 of file system.cpp.

**6.64.3.16 void mknix::System::calcInternalHeat ( )**

Definition at line 208 of file system.cpp.

**6.64.3.17 void mknix::System::calcMassMatrix ( )**

Definition at line 305 of file system.cpp.

**6.64.3.18 void mknix::System::calcTangentMatrix ( )**

Definition at line 352 of file system.cpp.

**6.64.3.19 void mknix::System::calcThermalTangentMatrix ( )**

Definition at line 220 of file system.cpp.

**6.64.3.20 bool mknix::System::checkAugmented ( )**

Definition at line 541 of file system.cpp.

**6.64.3.21 void mknix::System::clearAugmented ( )**

Definition at line 560 of file system.cpp.

**6.64.3.22 virtual Node\* mknix::System::getNode ( int *index* ) [inline], [virtual]**

Reimplemented in [mknix::SystemChain](#), and [mknix::SystemChain](#).

Definition at line 66 of file system.h.

**6.64.3.23 void mknix::System::getOutputSignalThermal ( double \* *vector\_in* )**

Definition at line 87 of file system.cpp.

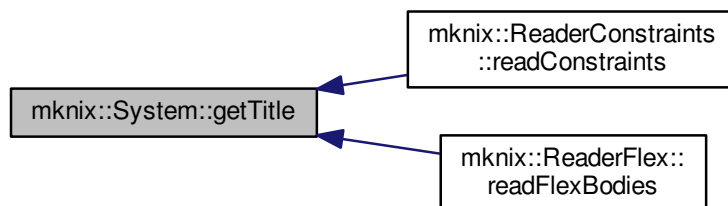
**6.64.3.24 void mknix::System::getThermalNodes ( std::vector< double > & *x\_coordinates* )**

Definition at line 80 of file system.cpp.

**6.64.3.25 std::string mknix::System::getTitle ( ) [inline]**

Definition at line 61 of file system.h.

Here is the caller graph for this function:



#### 6.64.3.26 void mknix::System::initFlexBodies ( )

Definition at line 123 of file system.cpp.

Here is the caller graph for this function:



#### 6.64.3.27 void mknix::System::outputStep ( const Imx::Vector< data\_type > & q, const Imx::Vector< data\_type > & qdot )

Definition at line 477 of file system.cpp.

#### 6.64.3.28 void mknix::System::outputStep ( const Imx::Vector< data\_type > & q )

Definition at line 497 of file system.cpp.

#### 6.64.3.29 void mknix::System::outputToFile ( std::ofstream \* outFile )

Definition at line 517 of file system.cpp.

#### 6.64.3.30 void mknix::System::setMechanical ( )

Definition at line 462 of file system.cpp.

#### 6.64.3.31 void mknix::System::update ( double time ) [virtual]

Reimplemented in [mknix::SystemChain](#), [mknix::SystemChain](#), and [mknix::Motion](#).

Definition at line 112 of file system.cpp.

#### 6.64.3.32 void mknix::System::updateThermalLoads ( double \* vector\_in )

Definition at line 103 of file system.cpp.

6.64.3.33 void mknix::System::writeBoundaryConnectivity ( std::vector< std::vector< **Point** \* > > & *connectivity\_nodes* )

Definition at line 588 of file system.cpp.

6.64.3.34 void mknix::System::writeBoundaryNodes ( std::vector< **Point** \* > & *boundary\_nodes* )

Definition at line 572 of file system.cpp.

6.64.3.35 void mknix::System::writeFlexBodies ( std::ofstream \* *outFile* )

Definition at line 142 of file system.cpp.

6.64.3.36 void mknix::System::writeJoints ( std::ofstream \* *outFile* )

Definition at line 153 of file system.cpp.

6.64.3.37 void mknix::System::writeRigidBodies ( std::ofstream \* *outFile* )

Definition at line 131 of file system.cpp.

#### 6.64.4 Friends And Related Function Documentation

6.64.4.1 friend class **Contact** [friend]

Definition at line 50 of file system.h.

6.64.4.2 friend class **Reader** [friend]

Definition at line 46 of file system.h.

6.64.4.3 friend class **ReaderConstraints** [friend]

Definition at line 49 of file system.h.

6.64.4.4 friend class **ReaderFlex** [friend]

Definition at line 47 of file system.h.

6.64.4.5 friend class **ReaderRigid** [friend]

Definition at line 48 of file system.h.

#### 6.64.5 Member Data Documentation

6.64.5.1 std::map< std::string, **Constraint**\* > mknix::System::constraints [protected]

Definition at line 149 of file system.h.

6.64.5.2 std::map< std::string, **ConstraintThermal**\* > mknix::System::constraintsThermal [protected]

Definition at line 150 of file system.h.

6.64.5.3 std::map< std::string, **FlexBody**\* > mknix::System::flexBodies [protected]

Definition at line 147 of file system.h.

6.64.5.4 std::vector<**Node**\*> mknix::System::groundNodes [protected]

Definition at line 143 of file system.h.

6.64.5.5 `std::vector< Load* > mknix::System::loads` [protected]

Definition at line 151 of file system.h.

6.64.5.6 `std::vector< LoadThermal* > mknix::System::loadsThermal` [protected]

Definition at line 152 of file system.h.

6.64.5.7 `std::vector< Motion* > mknix::System::motions` [protected]

Definition at line 154 of file system.h.

6.64.5.8 `bool mknix::System::outputMaxInterfaceTemp`

Definition at line 59 of file system.h.

6.64.5.9 `std::vector< Node* > mknix::System::outputSignalThermal` [protected]

Definition at line 153 of file system.h.

6.64.5.10 `std::map< std::string, RigidBody* > mknix::System::rigidBodies` [protected]

Definition at line 146 of file system.h.

6.64.5.11 `std::map< std::string, System* > mknix::System::subSystems` [protected]

Definition at line 145 of file system.h.

6.64.5.12 `std::map< std::string, Body* > mknix::System::thermalBodies` [protected]

Definition at line 148 of file system.h.

6.64.5.13 `std::string mknix::System::title` [protected]

Definition at line 142 of file system.h.

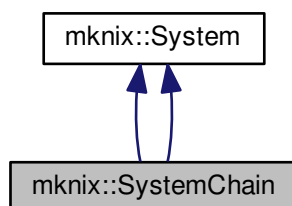
The documentation for this class was generated from the following files:

- [system.h](#)
- [system.cpp](#)

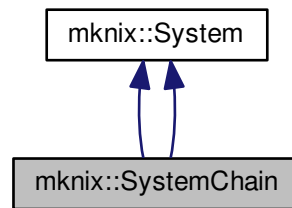
## 6.65 mknix::SystemChain Class Reference

```
#include <systemchain.h>
```

Inheritance diagram for mknix::SystemChain:



Collaboration diagram for mknix::SystemChain:



#### Public Member Functions

- [SystemChain](#) ()
- [SystemChain](#) (const char \*)
- [~SystemChain](#) ()
- void [setInterfaceNodeA](#) (double x, double y, double z)
- void [setInterfaceNodeB](#) (double x, double y, double z)
- void [setProperties](#) (int segments\_in, double length\_in)
- void [setMass](#) (double mass\_in)
- void [setTimeLengths](#) (std::map< double, double > &timeLengths\_in)
- [Node](#) \* [getNode](#) (int)
- void [addTimeLenght](#) (double time\_in, double length\_in)
- void [populate](#) ([Simulation](#) \*, std::string &)
- void [update](#) (double)
- [SystemChain](#) ()
- [SystemChain](#) (const char \*)
- [~SystemChain](#) ()
- void [setInterfaceNodeA](#) (double x, double y, double z)
- void [setInterfaceNodeB](#) (double x, double y, double z)
- void [setProperties](#) (int segments\_in, double length\_in)
- void [setMass](#) (double mass\_in)
- void [setTimeLengths](#) (std::map< double, double > &timeLengths\_in)
- [Node](#) \* [getNode](#) (int)
- void [addTimeLenght](#) (double time\_in, double length\_in)
- void [populate](#) ([Simulation](#) \*, std::string &)
- void [update](#) (double)

#### Additional Inherited Members

##### 6.65.1 Detailed Description

#### Author

AUTHORS <MAILS>

Definition at line 34 of file systemchain.h.

## 6.65.2 Constructor & Destructor Documentation

### 6.65.2.1 `mknix::SystemChain::SystemChain ( )`

Definition at line 30 of file `systemchain.cpp`.

### 6.65.2.2 `mknix::SystemChain::SystemChain ( const char * title_in )`

Definition at line 35 of file `systemchain.cpp`.

### 6.65.2.3 `mknix::SystemChain::~~SystemChain ( )`

Definition at line 40 of file `systemchain.cpp`.

### 6.65.2.4 `mknix::SystemChain::SystemChain ( )`

### 6.65.2.5 `mknix::SystemChain::SystemChain ( const char * )`

### 6.65.2.6 `mknix::SystemChain::~~SystemChain ( )`

## 6.65.3 Member Function Documentation

### 6.65.3.1 `void mknix::SystemChain::addTimeLenght ( double time_in, double length_in ) [inline]`

Definition at line 82 of file `systemchain.h`.

### 6.65.3.2 `void mknix::SystemChain::addTimeLenght ( double time_in, double length_in ) [inline]`

Definition at line 82 of file `systemchain2.h`.

### 6.65.3.3 `Node * mknix::SystemChain::getNode ( int node_i ) [virtual]`

Reimplemented from [mknix::System](#).

Definition at line 44 of file `systemchain.cpp`.

### 6.65.3.4 `Node* mknix::SystemChain::getNode ( int ) [virtual]`

Reimplemented from [mknix::System](#).

### 6.65.3.5 `void mknix::SystemChain::populate ( Simulation * theSimulation, std::string & energyKeyword )`

Definition at line 56 of file `systemchain.cpp`.

### 6.65.3.6 `void mknix::SystemChain::populate ( Simulation * , std::string & )`

### 6.65.3.7 `void mknix::SystemChain::setInterfaceNodeA ( double x, double y, double z ) [inline]`

Definition at line 49 of file `systemchain2.h`.

### 6.65.3.8 `void mknix::SystemChain::setInterfaceNodeA ( double x, double y, double z ) [inline]`

Definition at line 49 of file `systemchain.h`.

### 6.65.3.9 `void mknix::SystemChain::setInterfaceNodeB ( double x, double y, double z ) [inline]`

Definition at line 56 of file `systemchain.h`.

### 6.65.3.10 `void mknix::SystemChain::setInterfaceNodeB ( double x, double y, double z ) [inline]`

Definition at line 56 of file `systemchain2.h`.



6.65.3.11 void mknix::SystemChain::setMass ( double *mass\_in* ) [inline]

Definition at line 69 of file systemchain2.h.

6.65.3.12 void mknix::SystemChain::setMass ( double *mass\_in* ) [inline]

Definition at line 69 of file systemchain.h.

6.65.3.13 void mknix::SystemChain::setProperties ( int *segments\_in*, double *length\_in* ) [inline]

Definition at line 63 of file systemchain.h.

6.65.3.14 void mknix::SystemChain::setProperties ( int *segments\_in*, double *length\_in* ) [inline]

Definition at line 63 of file systemchain2.h.

6.65.3.15 void mknix::SystemChain::setTimeLengths ( std::map< double, double > & *timeLengths\_in* ) [inline]

Definition at line 75 of file systemchain2.h.

6.65.3.16 void mknix::SystemChain::setTimeLengths ( std::map< double, double > & *timeLengths\_in* ) [inline]

Definition at line 75 of file systemchain.h.

6.65.3.17 void mknix::SystemChain::update ( double ) [virtual]

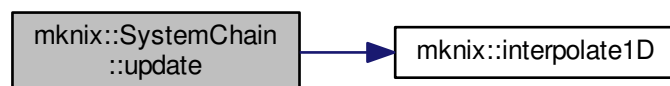
Reimplemented from [mknix::System](#).

6.65.3.18 void mknix::SystemChain::update ( double *theTime* ) [virtual]

Reimplemented from [mknix::System](#).

Definition at line 116 of file systemchain.cpp.

Here is the call graph for this function:



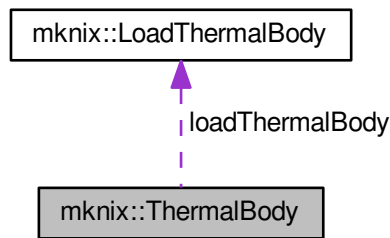
The documentation for this class was generated from the following files:

- [systemchain.h](#)
- [systemchain2.h](#)
- [systemchain.cpp](#)
- [systemchain2.cpp](#)

## 6.66 mknix::ThermalBody Class Reference

```
#include <bodythermal.h>
```

Collaboration diagram for mknix::ThermalBody:



### Public Member Functions

- [ThermalBody](#) ()
- [ThermalBody](#) (std::string)  
*Constructor with 1 parameter.*
- virtual [~ThermalBody](#) ()
- virtual void [initialize](#) ()  
*Cascade initialization funtion. Calls the initialize methods for each of the Cells and tells them to compute their shapefunction values. Both loops are parallelized.*
- virtual void [calcCapacityMatrix](#) ()  
*Computes the local Capacity of the material body by calling each cell's cascade function.*
- virtual void [calcConductivityMatrix](#) ()  
*Computes the local Conductivity of the material body by calling each cell's cascade function.*
- virtual void [calcExternalHeat](#) ()  
*Computes the local volumetric heat vector of the material body by calling each cell's cascade function.*
- virtual void [assembleCapacityMatrix](#) (Imx::Matrix< [data\\_type](#) > &)  
*Assembles the local conductivity into the global matrix by calling each cell's cascade function.*
- virtual void [assembleConductivityMatrix](#) (Imx::Matrix< [data\\_type](#) > &)  
*Assembles the local conductivity into the global matrix by calling each cell's cascade function.*
- virtual void [assembleExternalHeat](#) (Imx::Vector< [data\\_type](#) > &)  
*Assembles the local volumetric heat into the global heat load vector by calling each cell's cascade function.*
- virtual void [setOutput](#) (std::string)  
*Activates a flag for output data at the end of the analysis.*
- virtual void [outputStep](#) (const Imx::Vector< [data\\_type](#) > &, const Imx::Vector< [data\\_type](#) > &)  
*Postprocess and store step results for dynamic analysis.*
- virtual void [outputStep](#) (const Imx::Vector< [data\\_type](#) > &)  
*Postprocess and store step results for static analysis.*
- virtual void [outputToFile](#) (std::ofstream \*)  
*Streams the data stored during the analysis to a file. The idea is that all thermalBodies will be linked to a solid body with the same name (title). The postprocessor will treat this values as if it was the same body, so it must take into account that it can need to read several ENERGY(\*) sections for the same body.*
- virtual void [addNode](#) (Node \*node\_in)
- virtual Node \* [getNode](#) (int node\_number)
- virtual Node \* [getLastNode](#) ()
- virtual void [addCell](#) (int num, Cell \*cell\_in)
- virtual void [setLoadThermal](#) (LoadThermalBody \*theLoad)

## Protected Attributes

- std::string [title](#)
- bool [computeEnergy](#)
- std::vector< [Node](#) \* > [nodes](#)
- std::map< int, [Cell](#) \* > [cells](#)
- std::vector< [Imx::Vector](#)< [data\\_type](#) > \* > [temperature](#)
- [LoadThermalBody](#) \* [loadThermalBody](#)

## 6.66.1 Detailed Description

## Author

AUTHORS <MAILS>

Definition at line 35 of file `bodythermal.h`.

## 6.66.2 Constructor &amp; Destructor Documentation

## 6.66.2.1 mknix::ThermalBody::ThermalBody ( )

Definition at line 28 of file `bodythermal.cpp`.

6.66.2.2 mknix::ThermalBody::ThermalBody ( std::string *title\_in* )

Constructor with 1 parameter.

## Parameters

<i>title_in</i>	Name of body in the system. Will be the same as the associated material body
-----------------	--

Definition at line 40 of file `bodythermal.cpp`.

## 6.66.2.3 mknix::ThermalBody::~ThermalBody ( ) [virtual]

Definition at line 48 of file `bodythermal.cpp`.

## 6.66.3 Member Function Documentation

6.66.3.1 virtual void mknix::ThermalBody::addCell ( int *num*, [Cell](#) \* *cell\_in* ) [inline], [virtual]

Definition at line 93 of file `bodythermal.h`.

6.66.3.2 virtual void mknix::ThermalBody::addNode ( [Node](#) \* *node\_in* ) [inline], [virtual]

Definition at line 78 of file `bodythermal.h`.

6.66.3.3 void mknix::ThermalBody::assembleCapacityMatrix ( [Imx::Matrix](#)< [data\\_type](#) > & *globalCapacity* ) [virtual]

Assembles the local conductivity into the global matrix by calling each cell's cascade function.

## Parameters

<i>globalCapacity</i>	Reference to the global matrix of the thermal simulation.
-----------------------	---

## Returns

void

Definition at line 141 of file `bodythermal.cpp`.

**6.66.3.4** `void mknix::ThermalBody::assembleConductivityMatrix ( Imx::Matrix< data_type > & globalConductivity )`  
`[virtual]`

Assembles the local conductivity into the global matrix by calling each cell's cascade function.

Parameters

<i>global↔ Conductivity</i>	Reference to the global matrix of the thermal simulation.
---------------------------------	---

Returns

`void`

Definition at line 158 of file bodythermal.cpp.

**6.66.3.5** `void mknix::ThermalBody::assembleExternalHeat ( Imx::Vector< data_type > & globalExternalHeat )`  
`[virtual]`

Assembles the local volumetric heat into the global heat load vector by calling each cell's cascade function.

Returns

`void`

Definition at line 174 of file bodythermal.cpp.

**6.66.3.6** `void mknix::ThermalBody::calcCapacityMatrix ( )` `[virtual]`

Computes the local Capacity of the material body by calling each cell's cascade function.

Returns

`void`

Definition at line 91 of file bodythermal.cpp.

**6.66.3.7** `void mknix::ThermalBody::calcConductivityMatrix ( )` `[virtual]`

Computes the local Conductivity of the material body by calling each cell's cascade function.

Returns

`void`

Definition at line 107 of file bodythermal.cpp.

**6.66.3.8** `void mknix::ThermalBody::calcExternalHeat ( )` `[virtual]`

Computes the local volumetric heat vector of the material body by calling each cell's cascade function.

Returns

`void`

Definition at line 123 of file bodythermal.cpp.

**6.66.3.9** `virtual Node* mknix::ThermalBody::getLastNode ( )` `[inline],[virtual]`

Definition at line 88 of file bodythermal.h.

6.66.3.10 `virtual Node* mknix::ThermalBody::getNode ( int node_number )` `[inline],[virtual]`

Definition at line 83 of file `bodythermal.h`.

6.66.3.11 `void mknix::ThermalBody::initialize ( )` `[virtual]`

Cascade initialization funtion. Calls the initialize methods for each of the Cells and tells them to compute their shapefunction values. Both loops are parallelized.

Returns

void

Definition at line 58 of file `bodythermal.cpp`.

6.66.3.12 `void mknix::ThermalBody::outputStep ( const Imx::Vector< data_type > & q, const Imx::Vector< data_type > & qdot )` `[virtual]`

Postprocess and store step results for dynamic analysis.

Parameters

<i>q</i>	Global configuration vector
<i>qdot</i>	Global configuration first derivative vector

Returns

void

Definition at line 250 of file `bodythermal.cpp`.

6.66.3.13 `void mknix::ThermalBody::outputStep ( const Imx::Vector< data_type > & q )` `[virtual]`

Postprocess and store step results for static analysis.

Parameters

<i>q</i>	Global configuration vector
----------	-----------------------------

Returns

void

Definition at line 278 of file `bodythermal.cpp`.

6.66.3.14 `void mknix::ThermalBody::outputToFile ( std::ofstream * outFile )` `[virtual]`

Streams the data stored during the analysis to a file. The idea is that all thermalBodies will be linked to a solid body with the same name (title). The postprocessor will treat this values as if it was the same body, so it must take into account that it can need to read several ENERGY(.\*?) sections for the same body.

Parameters

<i>outFile</i>	Output files
----------------	--------------

Returns

void

Definition at line 211 of file `bodythermal.cpp`.

**6.66.3.15** `virtual void mknix::ThermalBody::setLoadThermal ( LoadThermalBody * theLoad )` `[inline],`  
`[virtual]`

Definition at line 99 of file `bodythermal.h`.

**6.66.3.16** `void mknix::ThermalBody::setOutput ( std::string outputType_in )` `[virtual]`

Activates a flag for output data at the end of the analysis.

See also

[outputToFile\(\)](#)  
[outputStep\(\)](#)

Parameters

<code>outputType_in</code>	Keyword of the flag. Options are: [ENERGY]
----------------------------	--

Returns

`void`

Definition at line 194 of file `bodythermal.cpp`.

#### 6.66.4 Member Data Documentation

**6.66.4.1** `std::map<int,Cell*> mknix::ThermalBody::cells` `[protected]`

Map of integration cells.

Definition at line 109 of file `bodythermal.h`.

**6.66.4.2** `bool mknix::ThermalBody::computeEnergy` `[protected]`

Definition at line 107 of file `bodythermal.h`.

**6.66.4.3** `LoadThermalBody* mknix::ThermalBody::loadThermalBody` `[protected]`

Definition at line 111 of file `bodythermal.h`.

**6.66.4.4** `std::vector<Node*> mknix::ThermalBody::nodes` `[protected]`

Definition at line 108 of file `bodythermal.h`.

**6.66.4.5** `std::vector<Imx::Vector<data_type>*> mknix::ThermalBody::temperature` `[protected]`

Definition at line 110 of file `bodythermal.h`.

**6.66.4.6** `std::string mknix::ThermalBody::title` `[protected]`

Definition at line 105 of file `bodythermal.h`.

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

- [bodythermal.h](#)
- [bodythermal.cpp](#)

#### 6.67 Imx::Vector< T > Class Template Reference

```
#include <cell.h>
```

### 6.67.1 Detailed Description

```
template<typename T>class Imx::Vector< T >
```

Definition at line 39 of file cell.h.

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

- [cell.h](#)

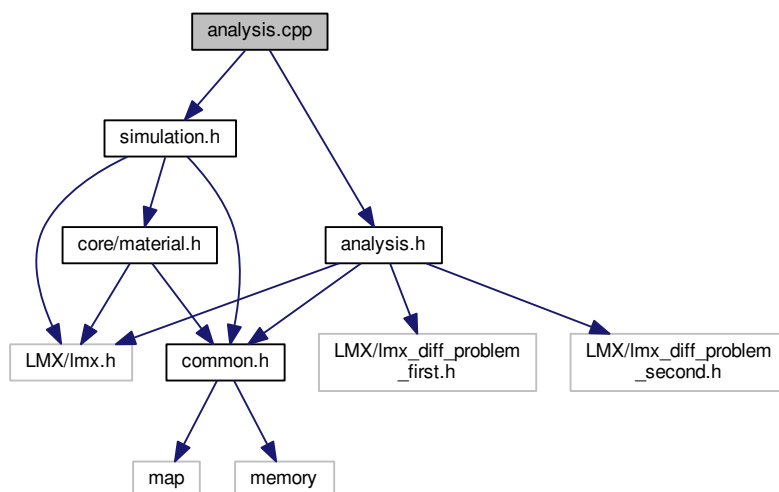
## 7 File Documentation

### 7.1 analysis.cpp File Reference

```
#include "analysis.h"
```

```
#include "simulation.h"
```

Include dependency graph for analysis.cpp:



### Namespaces

- [mknix](#)

### 7.2 analysis.h File Reference

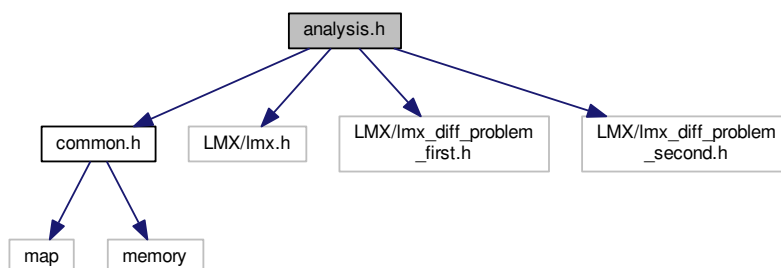
```
#include "common.h"
```

```
#include "LMX/lmx.h"
```

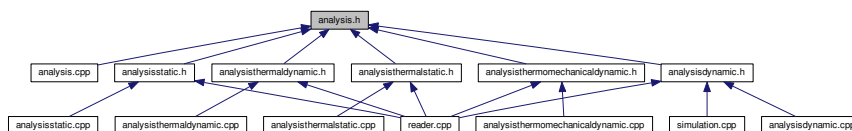
```
#include "LMX/lmx_diff_problem_first.h"
```

```
#include "LMX/lmx_diff_problem_second.h"
```

Include dependency graph for analysis.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `mnix::Analysis`

## Namespaces

- `mnix`

## 7.3 analysisdynamic.cpp File Reference

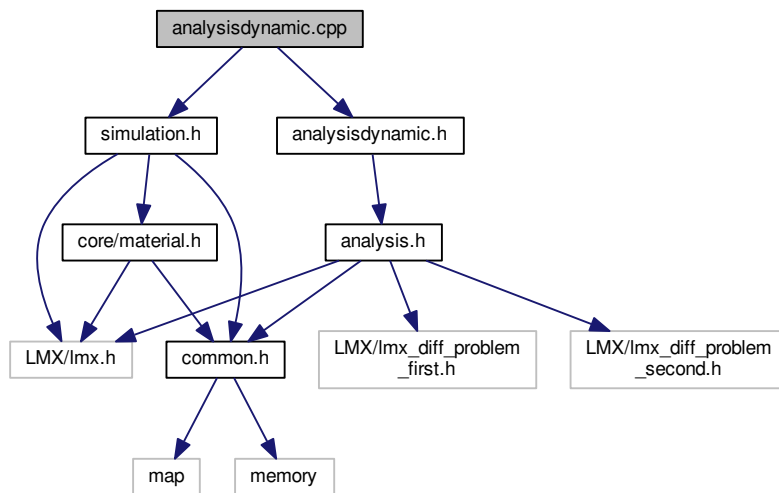
```

#include "analysisdynamic.h"
#include "simulation.h"

```



Include dependency graph for analysisdynamic.cpp:



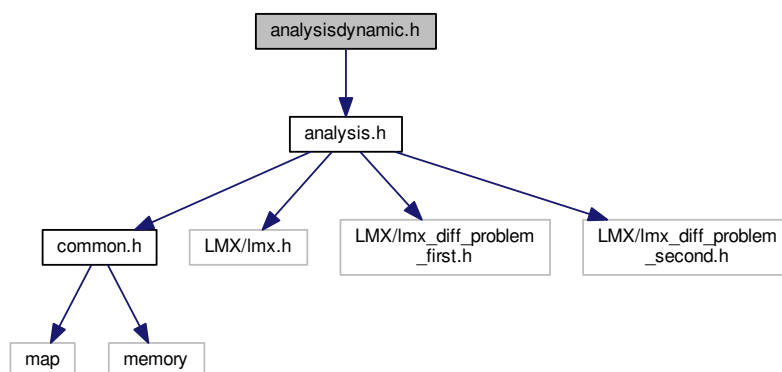
## Namespaces

- [mknix](#)

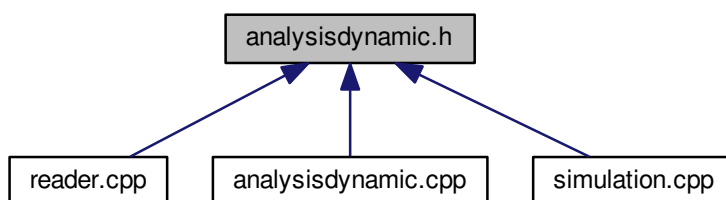
## 7.4 analysisdynamic.h File Reference

```
#include "analysis.h"
```

Include dependency graph for analysisdynamic.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::AnalysisDynamic`

#### Namespaces

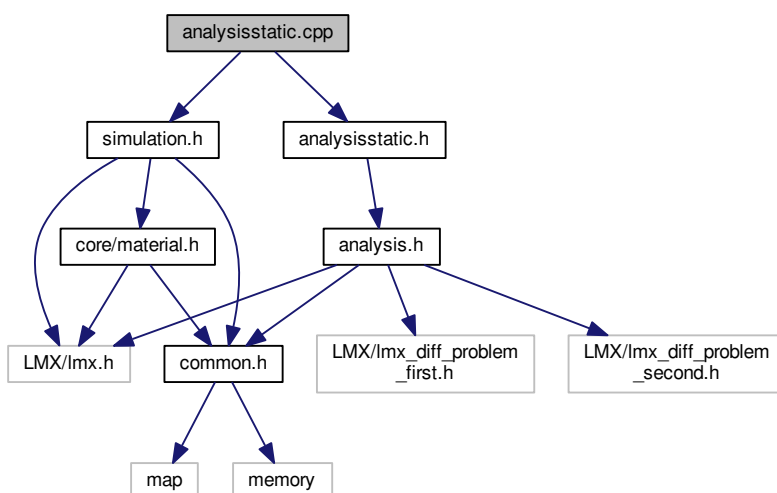
- `mnix`

### 7.5 analysisstatic.cpp File Reference

```
#include "analysisstatic.h"
```

```
#include "simulation.h"
```

Include dependency graph for `analysisstatic.cpp`:



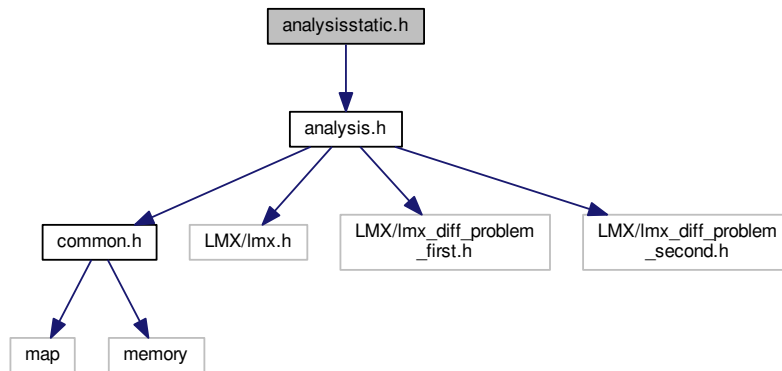
#### Namespaces

- `mnix`

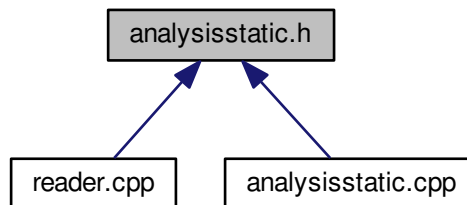
## 7.6 analysisstatic.h File Reference

```
#include "analysis.h"
```

Include dependency graph for analysisstatic.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [mknix::AnalysisStatic](#)

### Namespaces

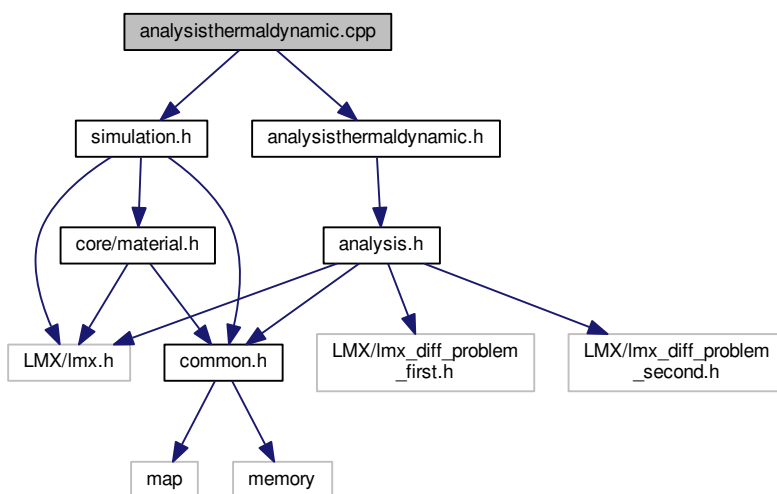
- [mknix](#)

## 7.7 analysisthermaldynamic.cpp File Reference

```
#include "analysisthermaldynamic.h"
```

```
#include "simulation.h"
```

Include dependency graph for `analysisthermaldynamic.cpp`:



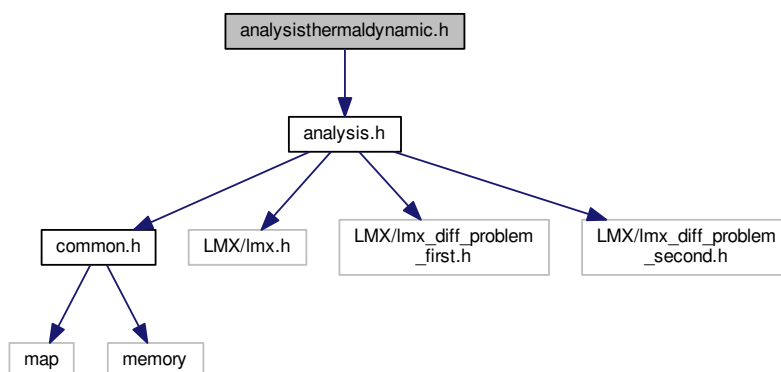
## Namespaces

- [mknix](#)

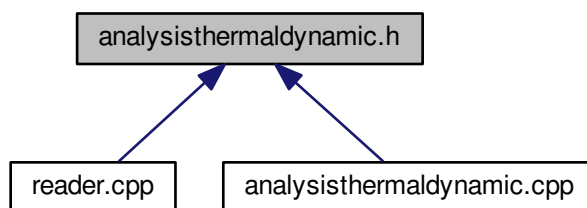
## 7.8 analysisthermaldynamic.h File Reference

```
#include "analysis.h"
```

Include dependency graph for `analysisthermaldynamic.h`:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mknix::AnalysisThermalDynamic`

#### Namespaces

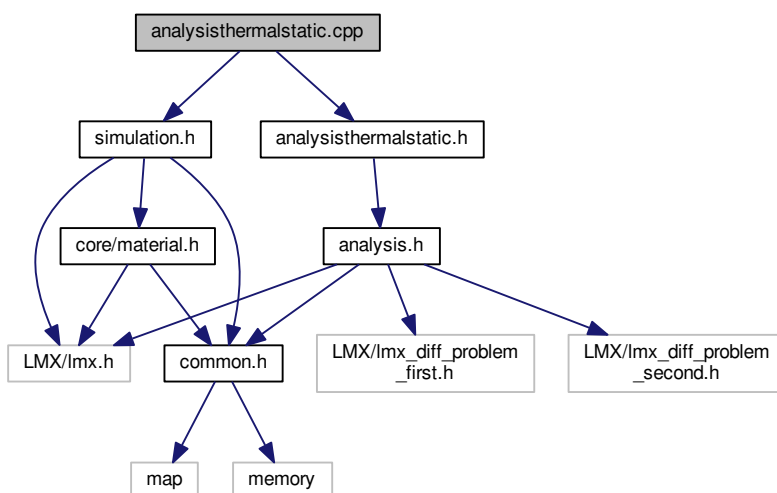
- `mknix`

## 7.9 analysisthermalstatic.cpp File Reference

```
#include "analysisthermalstatic.h"
```

```
#include "simulation.h"
```

Include dependency graph for `analysisthermalstatic.cpp`:



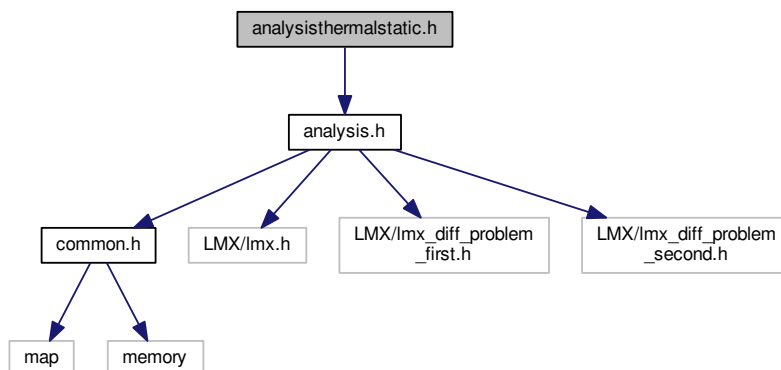
#### Namespaces

- `mknix`

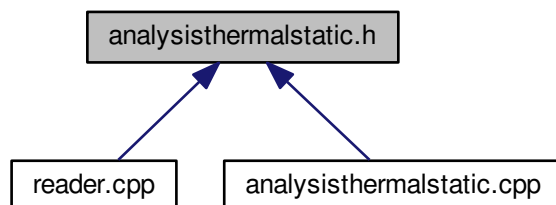
## 7.10 analysisthermalstatic.h File Reference

```
#include "analysis.h"
```

Include dependency graph for analysisthermalstatic.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class `mnix::AnalysisThermalStatic`

### Namespaces

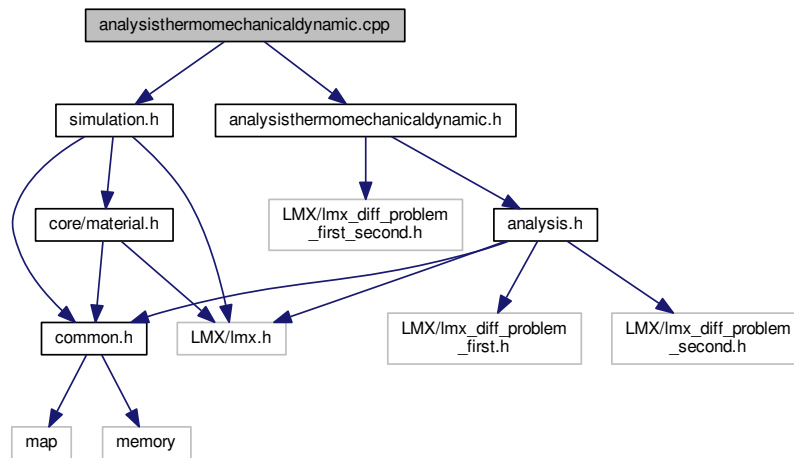
- `mnix`

## 7.11 analysisthermomechanicaldynamic.cpp File Reference

```
#include "analysisthermomechanicaldynamic.h"
```

```
#include "simulation.h"
```

Include dependency graph for analysisthermomechanicaldynamic.cpp:



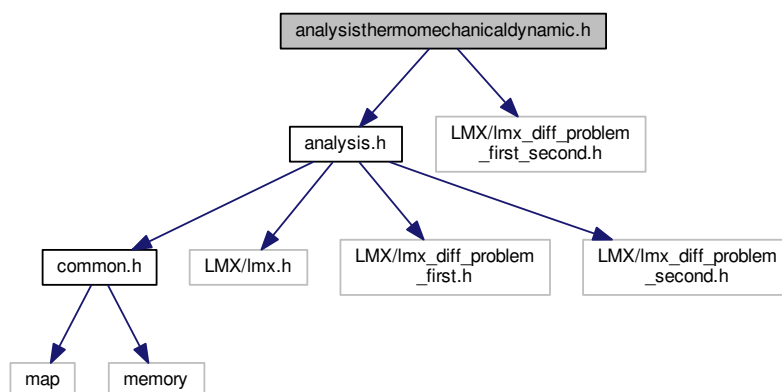
## Namespaces

- [mknix](#)

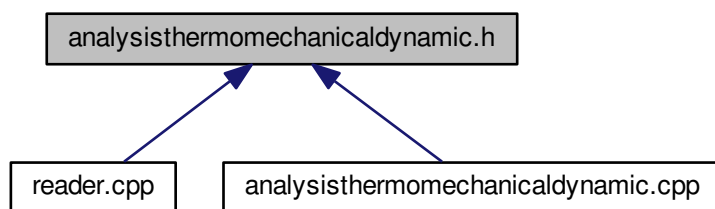
## 7.12 analysisthermomechanicaldynamic.h File Reference

```
#include "analysis.h"
#include <LMX/lmx_diff_problem_first_second.h>
```

Include dependency graph for analysisthermomechanicaldynamic.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::AnalysisThermoMechanicalDynamic`

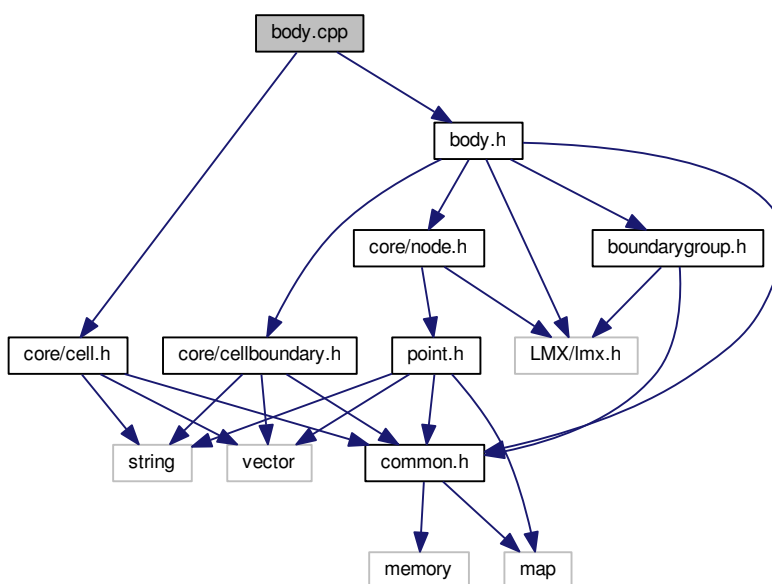
#### Namespaces

- `mnix`

### 7.13 body.cpp File Reference

```
#include "body.h"
#include <core/cell.h>
```

Include dependency graph for `body.cpp`:



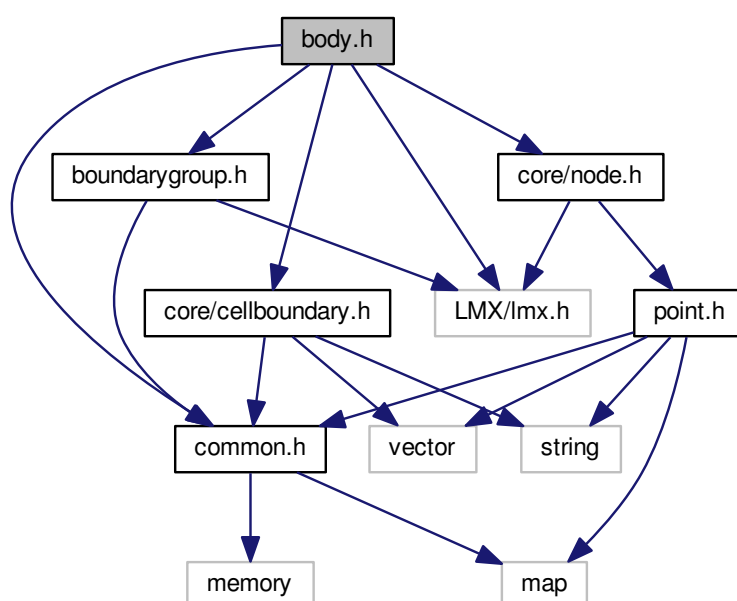


## Namespaces

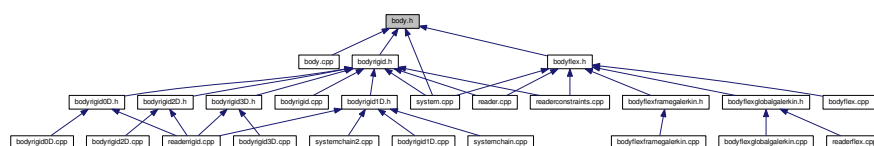
- `mknix`

## 7.14 body.h File Reference

```
#include "common.h"
#include "LMX/lmx.h"
#include "boundarygroup.h"
#include <core/cellboundary.h>
#include <core/node.h>
Include dependency graph for body.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

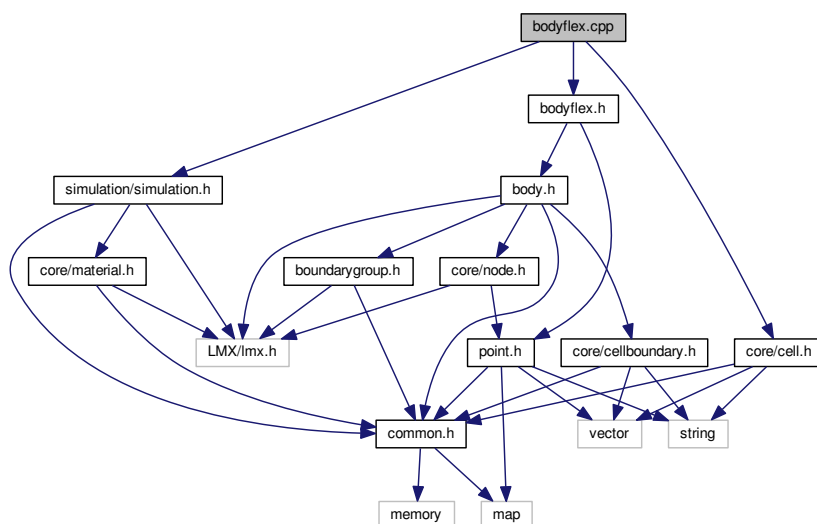
- class `mknix::Body`

## Namespaces

- [mknix](#)

## 7.15 bodyflex.cpp File Reference

```
#include "bodyflex.h"
#include <core/cell.h>
#include <simulation/simulation.h>
Include dependency graph for bodyflex.cpp:
```



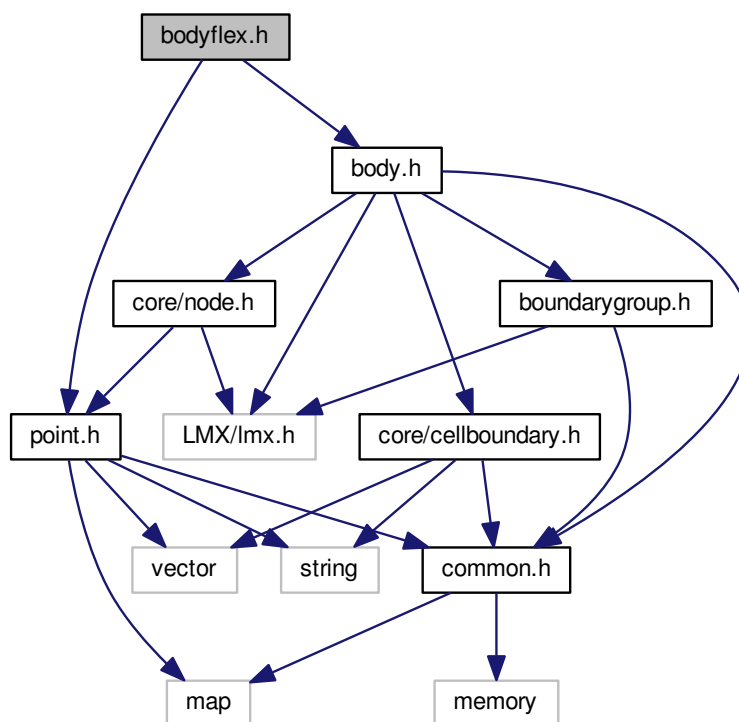
## Namespaces

- [mknix](#)

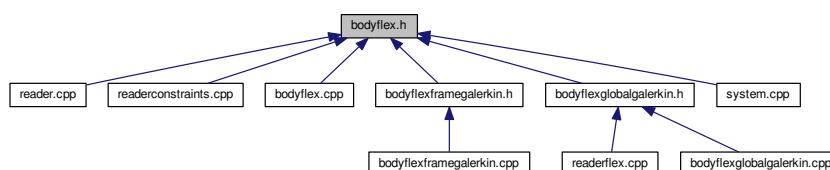
## 7.16 bodyflex.h File Reference

```
#include "body.h"
#include <core/point.h>
```

Include dependency graph for bodyflex.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mnix::FlexBody](#)

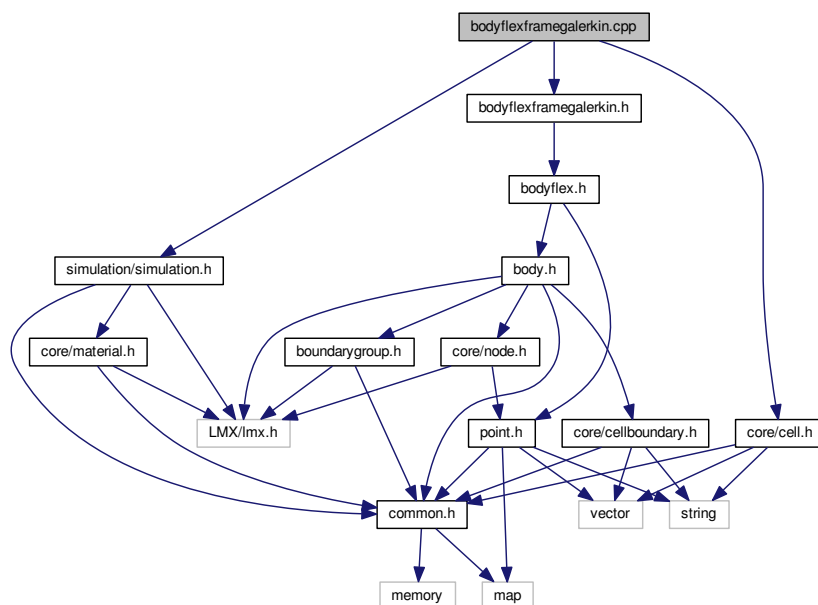
## Namespaces

- [mnix](#)

## 7.17 bodyflexframegalerkin.cpp File Reference

```
#include "bodyflexframegalerkin.h"
```

```
#include <core/cell.h>
#include <simulation/simulation.h>
Include dependency graph for bodyflexframegalerkin.cpp:
```



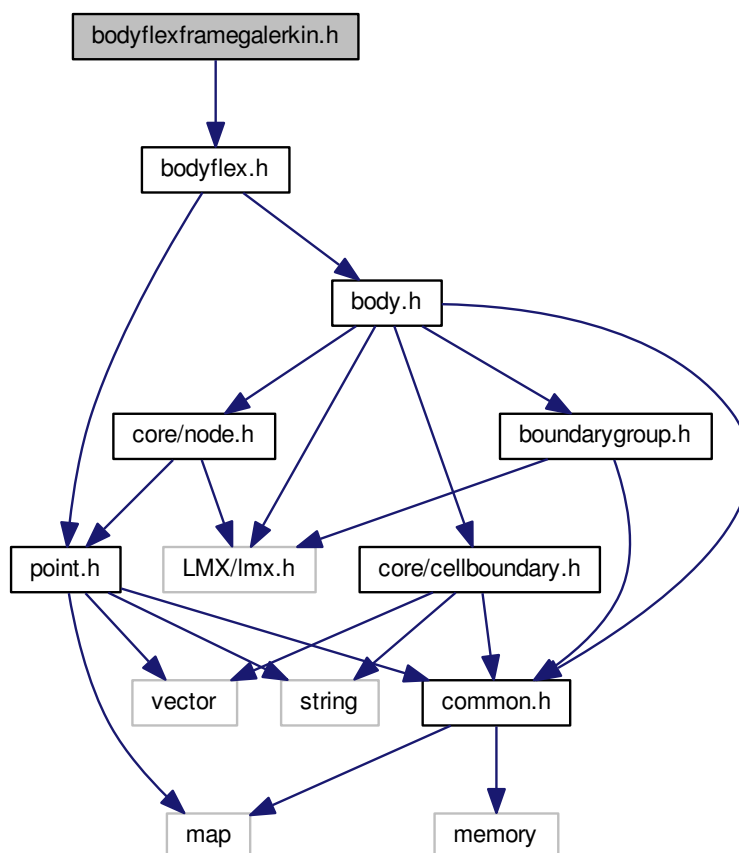
## Namespaces

- [mknix](#)

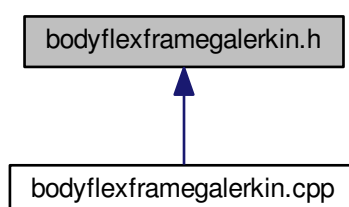
## 7.18 bodyflexframegalerkin.h File Reference

```
#include "bodyflex.h"
```

Include dependency graph for bodyflexframegalerkin.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mnix::FlexFrameGalerkin](#)

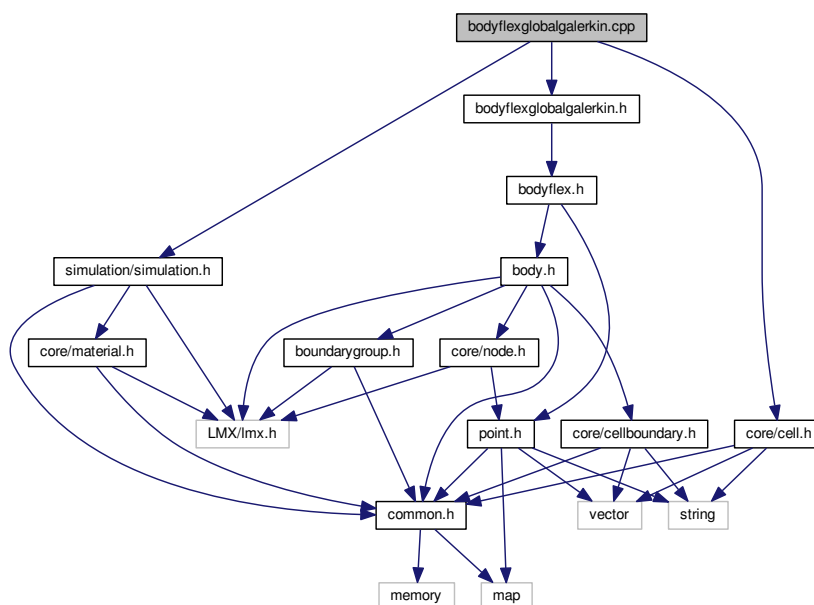
## Namespaces

- [mknix](#)

## 7.19 bodyflexglobalgalerkin.cpp File Reference

```
#include "bodyflexglobalgalerkin.h"
#include <core/cell.h>
#include <simulation/simulation.h>
```

Include dependency graph for bodyflexglobalgalerkin.cpp:



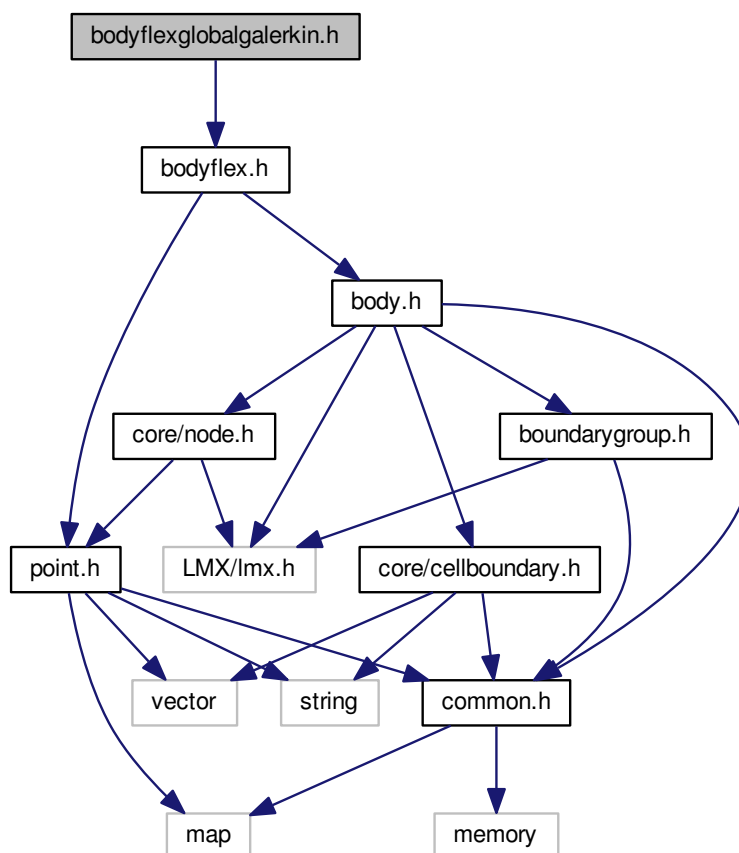
## Namespaces

- [mknix](#)

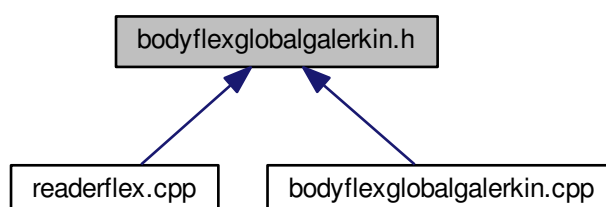
## 7.20 bodyflexglobalgalerkin.h File Reference

```
#include "bodyflex.h"
```

Include dependency graph for bodyflexglobalgalerkin.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mnix::FlexGlobalGalerkin](#)

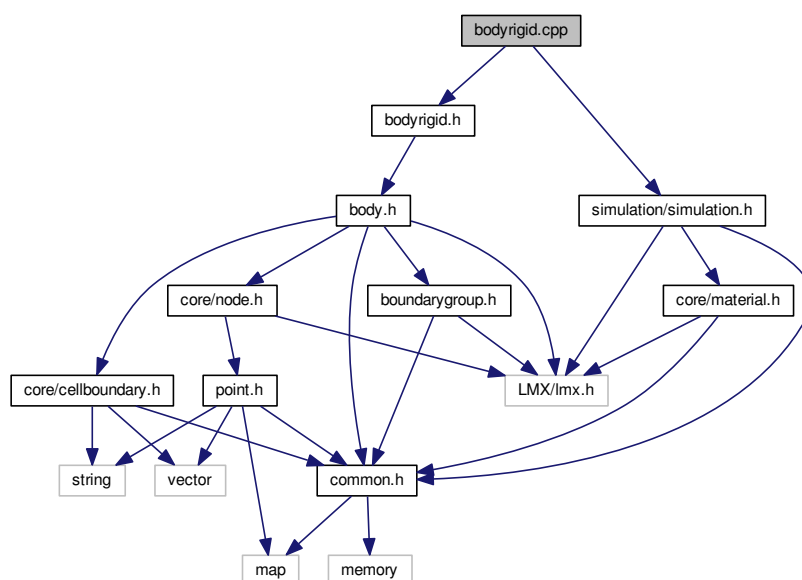
## Namespaces

- [mknix](#)

## 7.21 bodyrigid.cpp File Reference

```
#include "bodyrigid.h"
#include <simulation/simulation.h>
```

Include dependency graph for bodyrigid.cpp:



## Namespaces

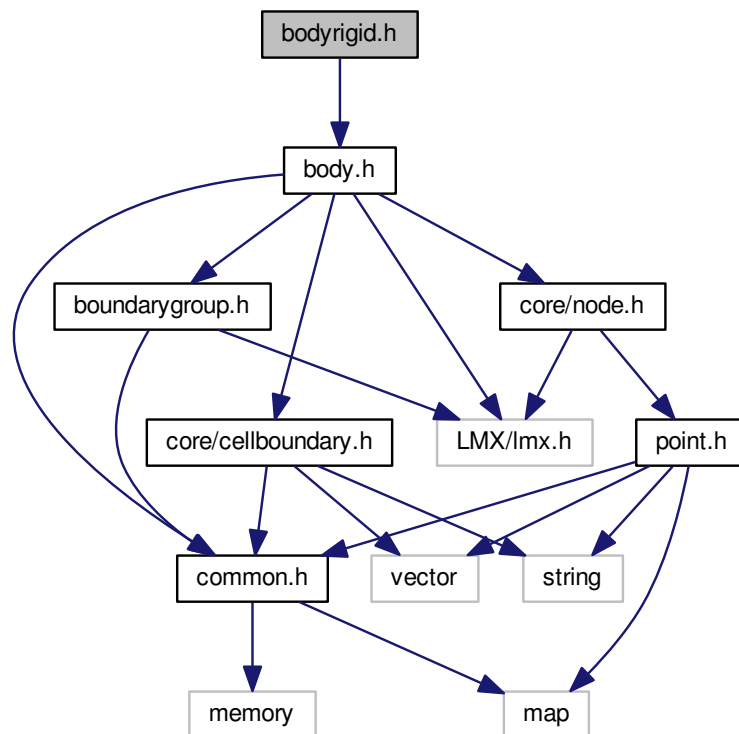
- [mknix](#)

## 7.22 bodyrigid.h File Reference

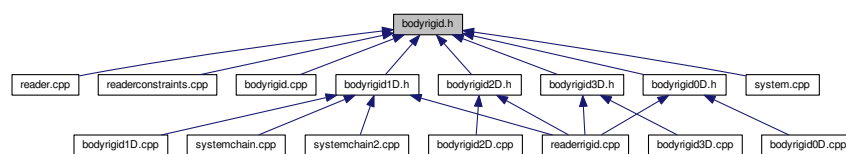
```
#include "body.h"
```



Include dependency graph for bodyrigid.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `mnix::RigidBody`

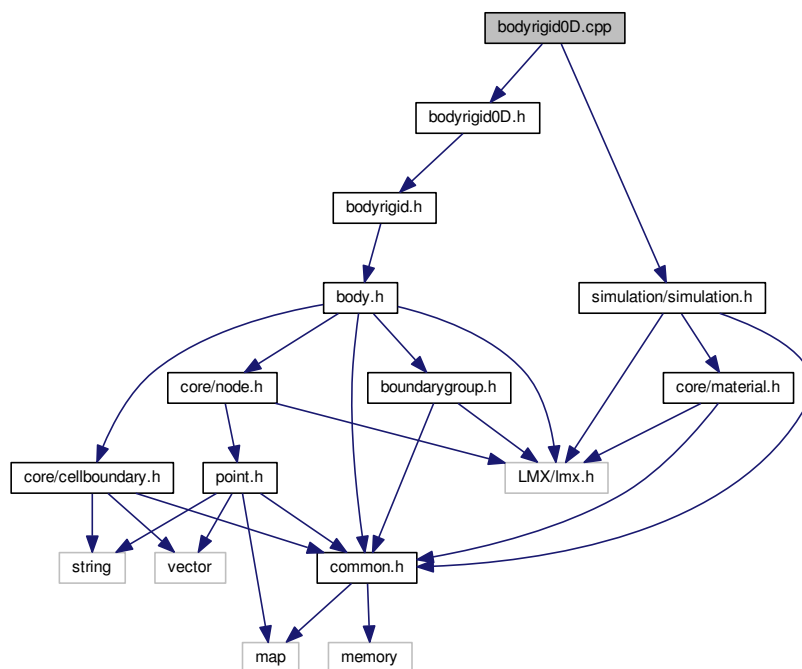
## Namespaces

- `mnix`

## 7.23 bodyrigid0D.cpp File Reference

```
#include "bodyrigid0D.h"
```

```
#include <simulation/simulation.h>
Include dependency graph for bodyrigid0D.cpp:
```



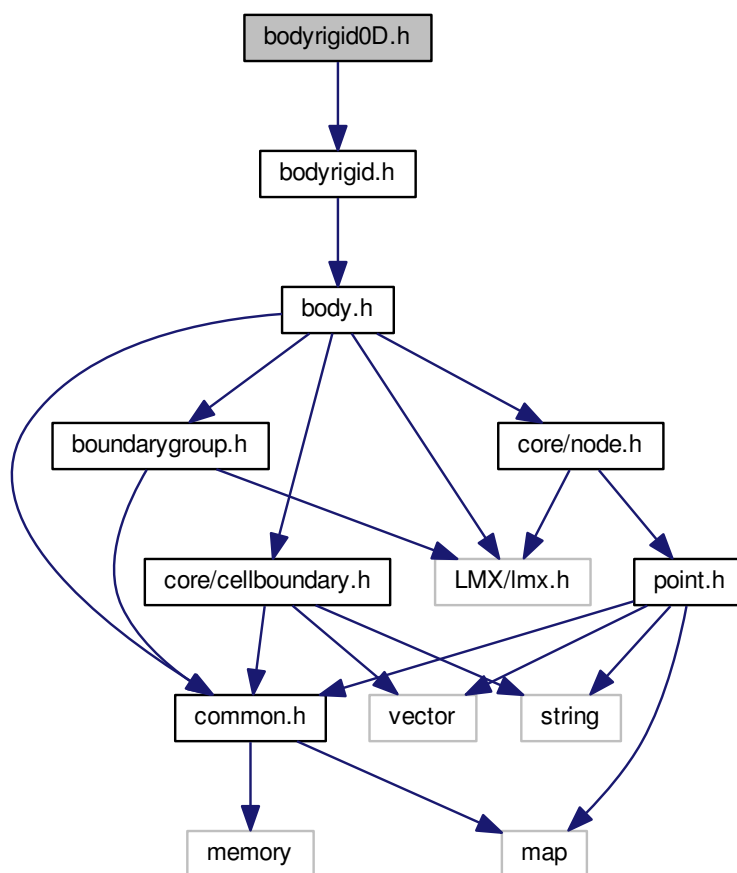
## Namespaces

- [mknix](#)

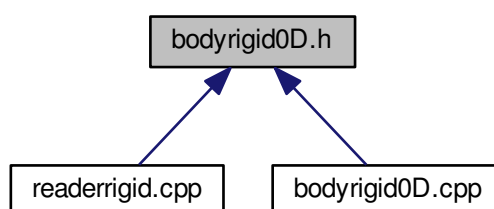
## 7.24 bodyrigid0D.h File Reference

```
#include "bodyrigid.h"
```

Include dependency graph for bodyrigid0D.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mnix::RigidBodyMassPoint](#)

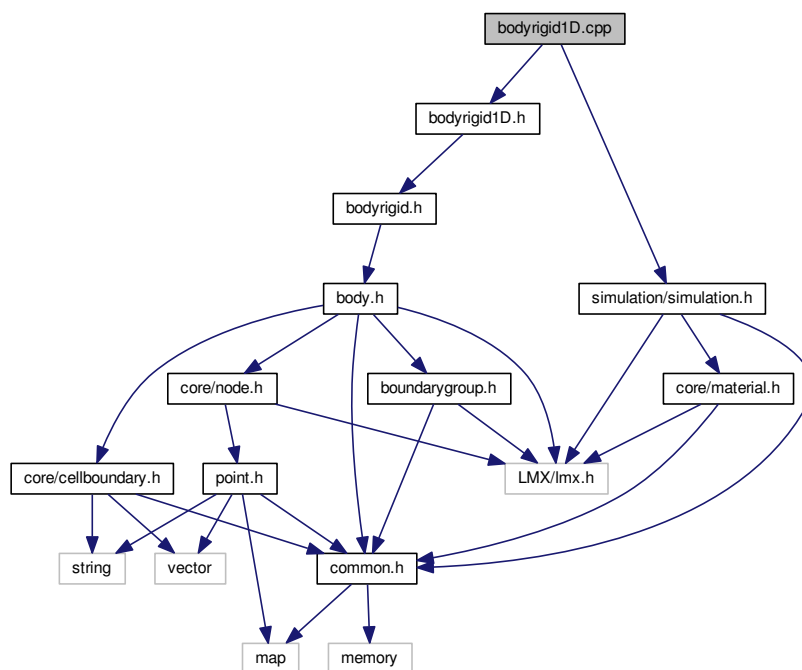
## Namespaces

- [mknix](#)

## 7.25 bodyrigid1D.cpp File Reference

```
#include "bodyrigid1D.h"
#include <simulation/simulation.h>
```

Include dependency graph for bodyrigid1D.cpp:



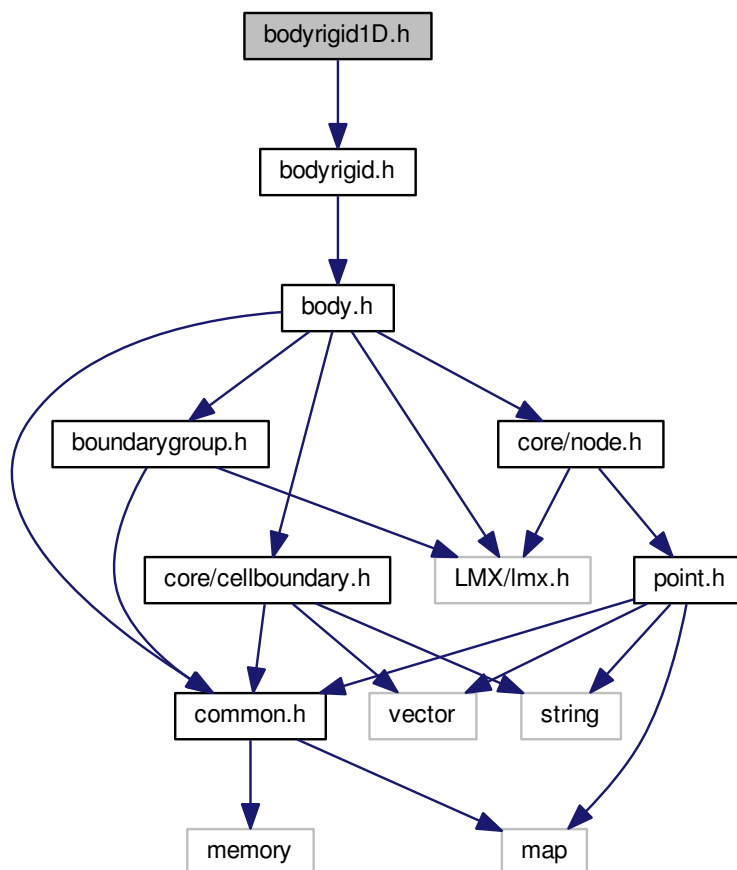
## Namespaces

- [mknix](#)

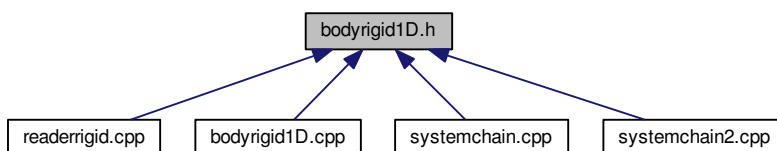
## 7.26 bodyrigid1D.h File Reference

```
#include "bodyrigid.h"
```

Include dependency graph for bodyrigid1D.h:



This graph shows which files directly or indirectly include this file:



#### Classes

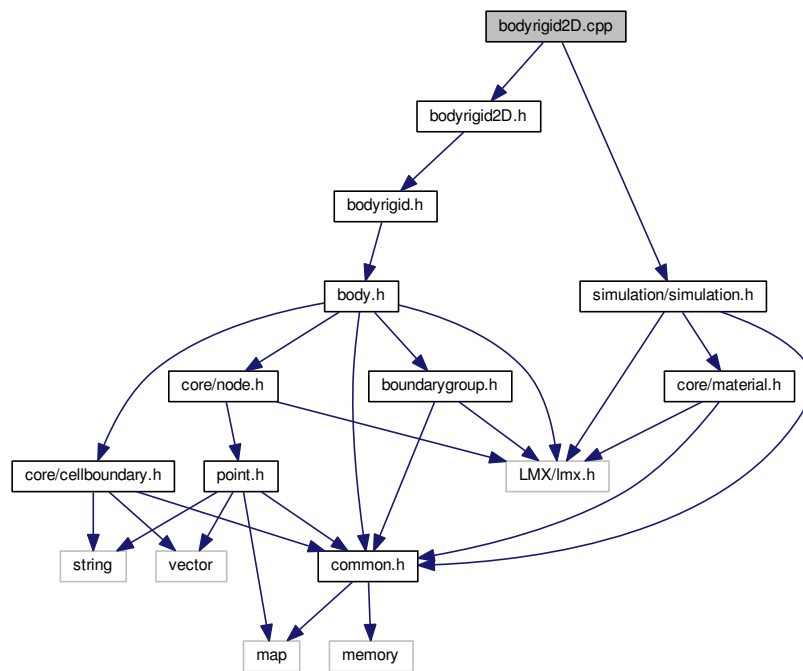
- class [mnix::RigidBar](#)

#### Namespaces

- [mnix](#)

## 7.27 bodyrigid2D.cpp File Reference

```
#include "bodyrigid2D.h"
#include <simulation/simulation.h>
Include dependency graph for bodyrigid2D.cpp:
```



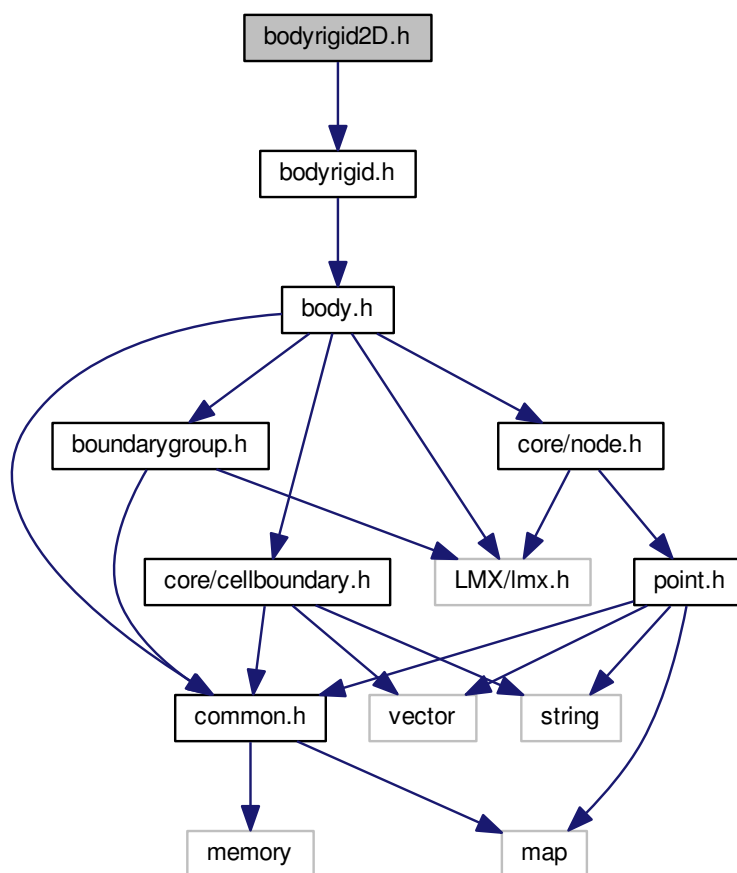
## Namespaces

- [mknix](#)

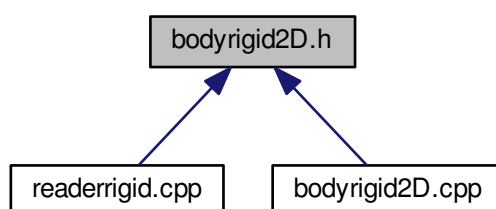
## 7.28 bodyrigid2D.h File Reference

```
#include "bodyrigid.h"
```

Include dependency graph for bodyrigid2D.h:



This graph shows which files directly or indirectly include this file:



#### Classes

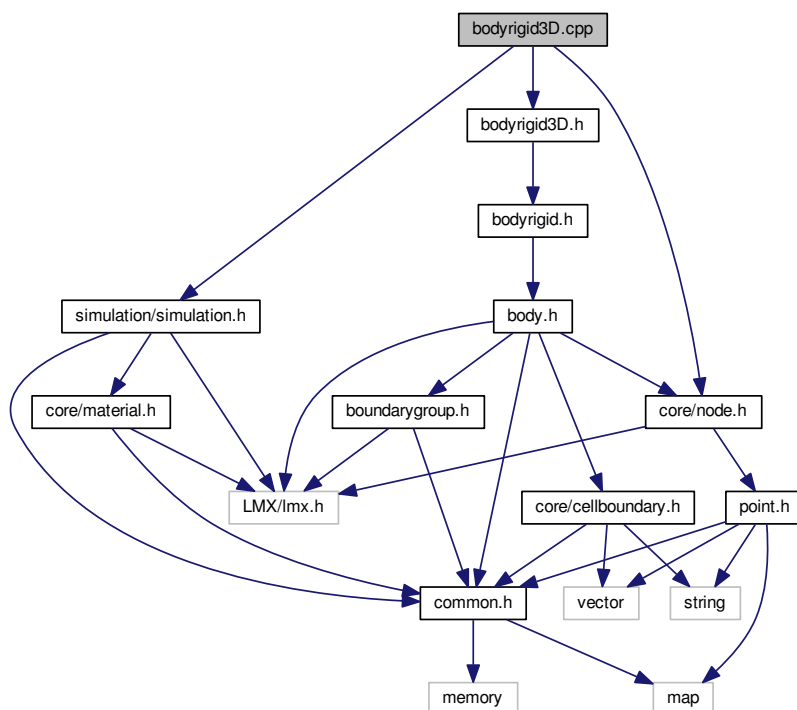
- class [mknix::RigidBody2D](#)

## Namespaces

- [mknix](#)

## 7.29 bodyrigid3D.cpp File Reference

```
#include "bodyrigid3D.h"
#include <core/node.h>
#include <simulation/simulation.h>
Include dependency graph for bodyrigid3D.cpp:
```



## Namespaces

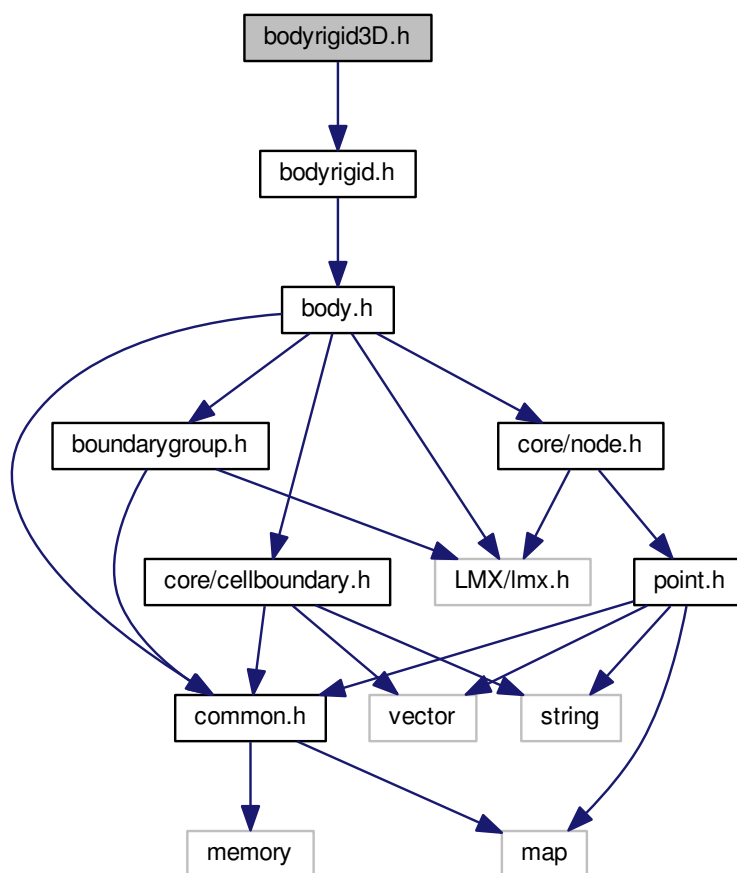
- [mknix](#)

## 7.30 bodyrigid3D.h File Reference

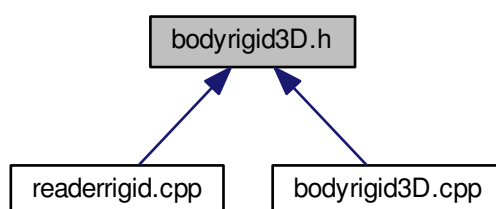
```
#include "bodyrigid.h"
```



Include dependency graph for bodyrigid3D.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::RigidBody3D](#)

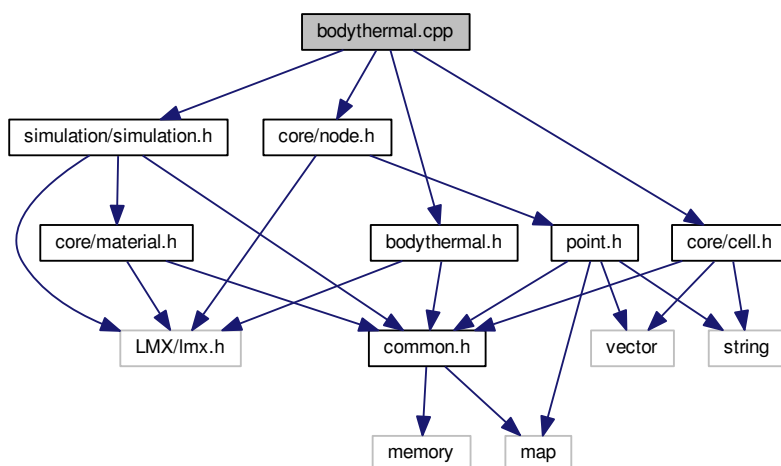
## Namespaces

- [mknix](#)

## 7.31 bodythermal.cpp File Reference

```
#include "bodythermal.h"
#include <core/node.h>
#include <core/cell.h>
#include <simulation/simulation.h>
```

Include dependency graph for bodythermal.cpp:



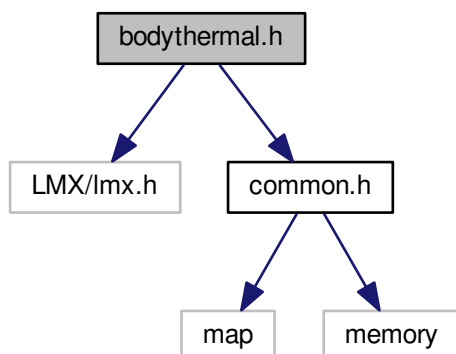
## Namespaces

- [mknix](#)

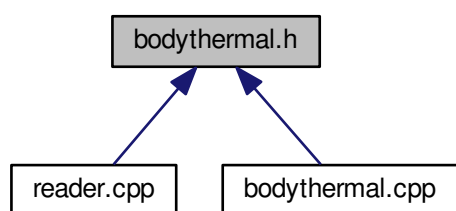
## 7.32 bodythermal.h File Reference

```
#include "LMX/lmx.h"
#include "common.h"
```

Include dependency graph for bodythermal.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::ThermalBody](#)

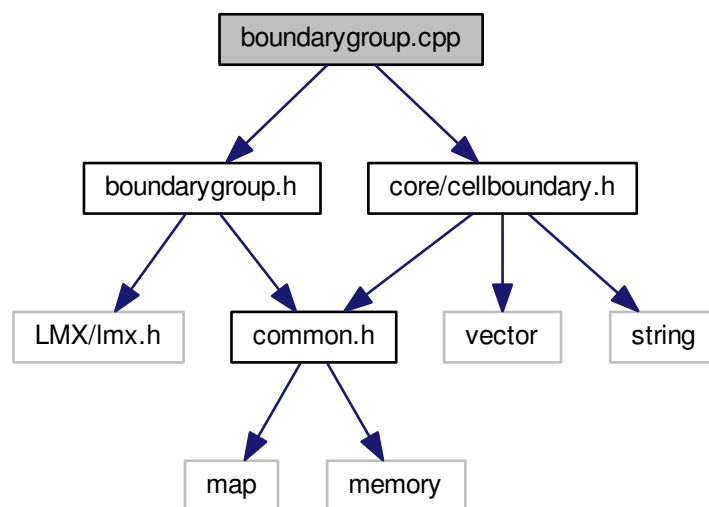
#### Namespaces

- [mknix](#)

### 7.33 boundarygroup.cpp File Reference

```
#include "boundarygroup.h"
#include <core/cellboundary.h>
```

Include dependency graph for boundarygroup.cpp:



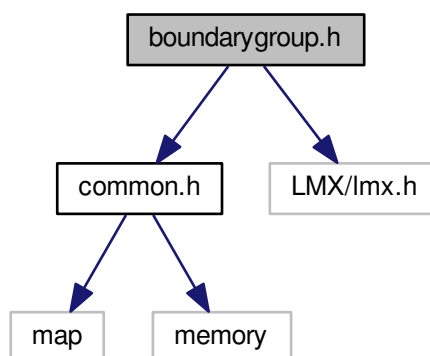
#### Namespaces

- [mknix](#)

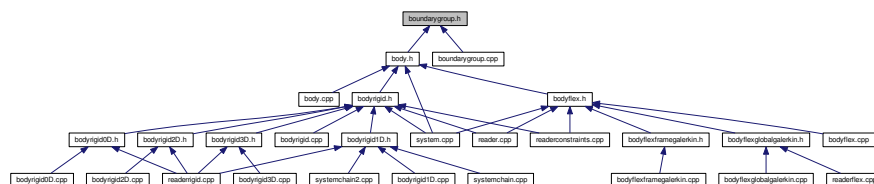
### 7.34 boundarygroup.h File Reference

```
#include "common.h"  
#include "LMX/lmx.h"
```

Include dependency graph for boundarygroup.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `mknix::BoundaryGroup`

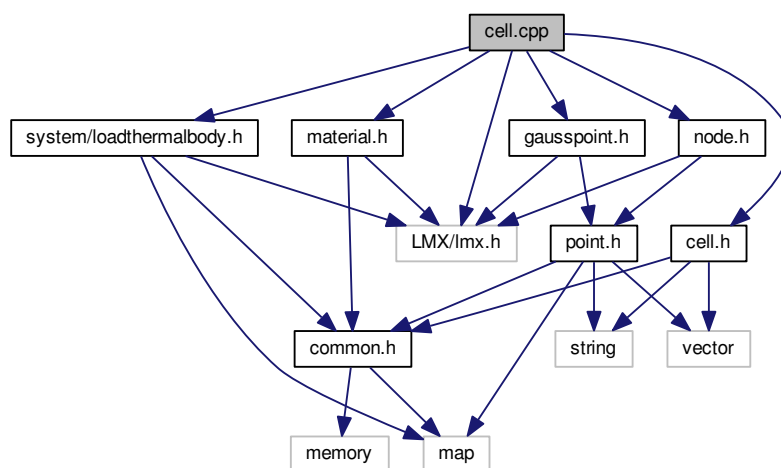
## Namespaces

- `mknix`

### 7.35 cell.cpp File Reference

```
#include "LMX/lmx.h"
#include "cell.h"
#include "material.h"
#include "gausspoint.h"
#include "node.h"
#include <system/loadthermalbody.h>
```

Include dependency graph for cell.cpp:



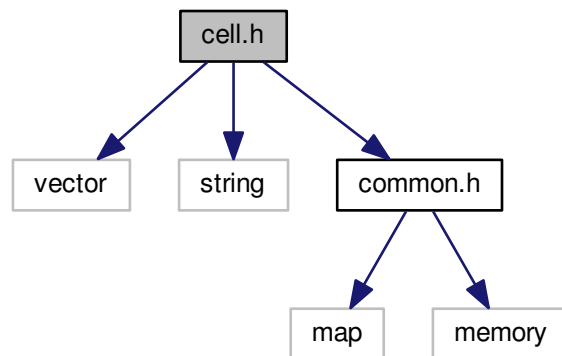
## Namespaces

- `mknix`

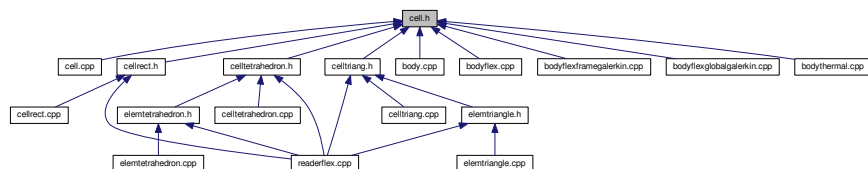
### 7.36 cell.h File Reference

Background cells for integration.

```
#include <vector>
#include <string>
#include <common.h>
Include dependency graph for cell.h:
```



This graph shows which files directly or indirectly include this file:



#### Classes

- class `Imx::Vector< T >`
- class `Imx::Matrix< T >`
- class `mnix::Cell`

#### Namespaces

- `Imx`
- `mnix`

#### 7.36.1 Detailed Description

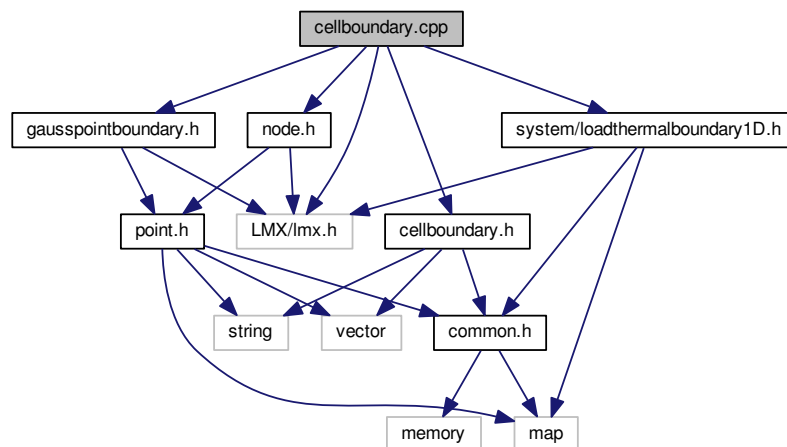
Background cells for integration.

Author

Daniel Iglesias

## 7.37 cellboundary.cpp File Reference

```
#include "LMX/lmx.h"  
#include "node.h"  
#include "cellboundary.h"  
#include "gausspointboundary.h"  
#include <system/loadthermalboundary1D.h>  
Include dependency graph for cellboundary.cpp:
```



Namespaces

- [mknix](#)

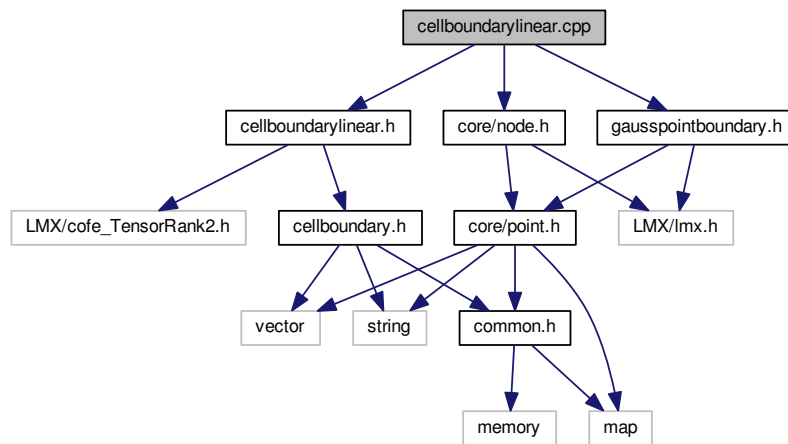
## 7.38 cellboundary.h File Reference

```
#include <vector>  
#include <string>  
#include "common.h"
```





Include dependency graph for cellboundarylinear.cpp:



#### Namespaces

- [mknix](#)

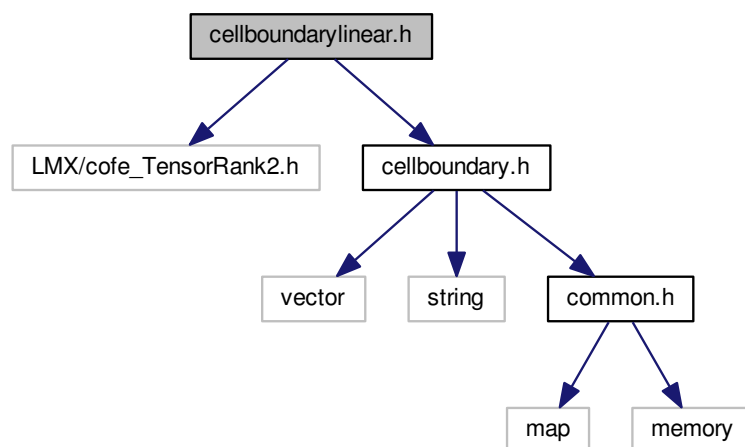
#### 7.40 cellboundarylinear.h File Reference

Background cells for boundary integration.

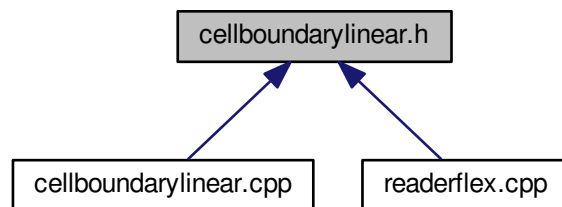
```
#include "LMX/cofe_TensorRank2.h"
```

```
#include "cellboundary.h"
```

Include dependency graph for cellboundarylinear.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::CellBoundaryLinear](#)

#### Namespaces

- [mknix](#)

#### 7.40.1 Detailed Description

Background cells for boundary integration.

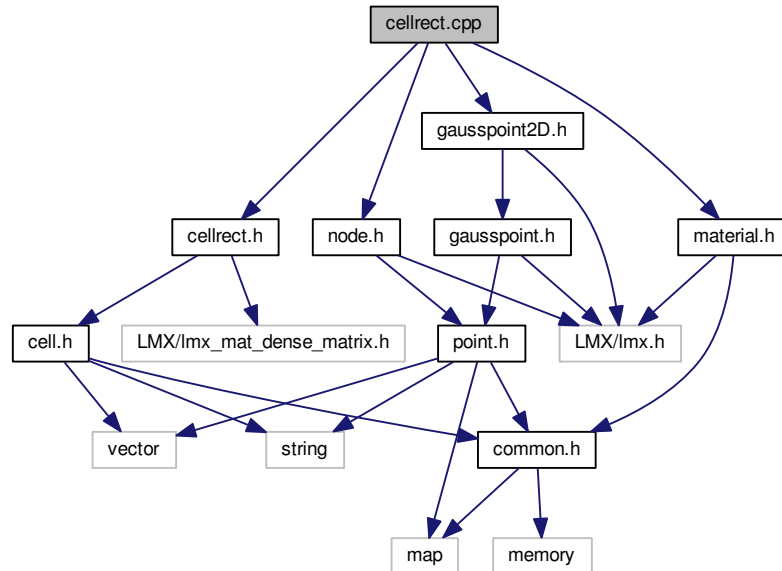
#### Author

Daniel Iglesias

#### 7.41 cellrect.cpp File Reference

```
#include "cellrect.h"
#include "material.h"
#include "node.h"
#include "gausspoint2D.h"
```

Include dependency graph for cellrect.cpp:



## Namespaces

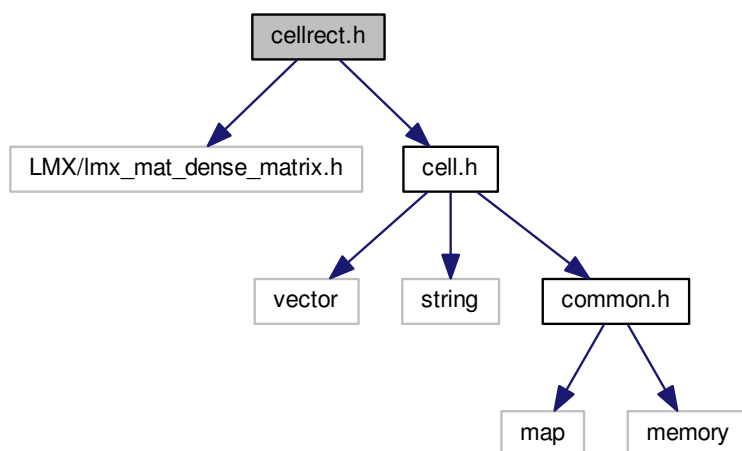
- [mknix](#)

## 7.42 cellrect.h File Reference

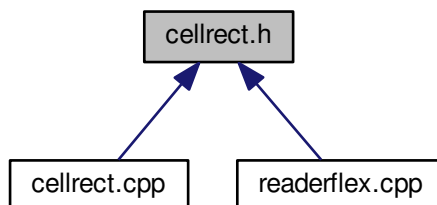
Background cells for integration.

```
#include "LMX/lmx_mat_dense_matrix.h"
#include "cell.h"
```

Include dependency graph for cellrect.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `lmx::DenseMatrix< T >`
- class `mnix::CellRect`

#### Namespaces

- `lmx`
- `mnix`

#### 7.42.1 Detailed Description

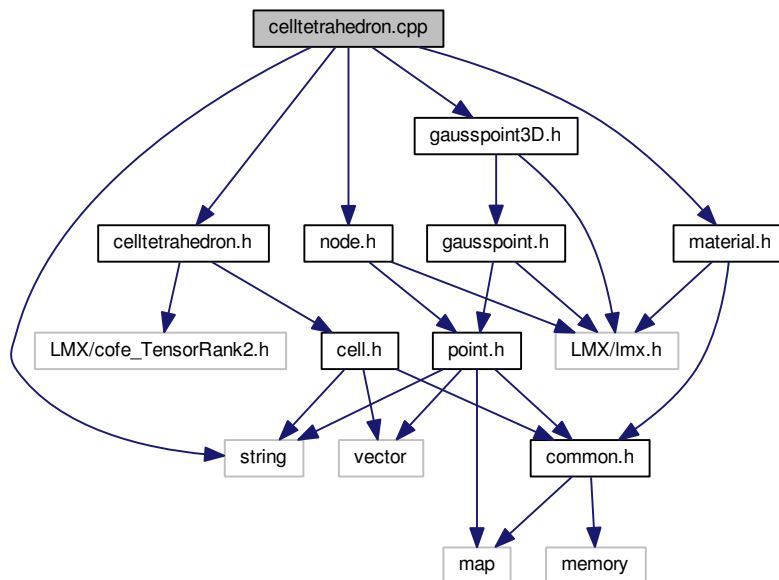
Background cells for integration.

Author

Daniel Iglesias

## 7.43 celltetrahedron.cpp File Reference

```
#include "celltetrahedron.h"
#include "material.h"
#include "node.h"
#include "gausspoint3D.h"
#include <string>
Include dependency graph for celltetrahedron.cpp:
```



Namespaces

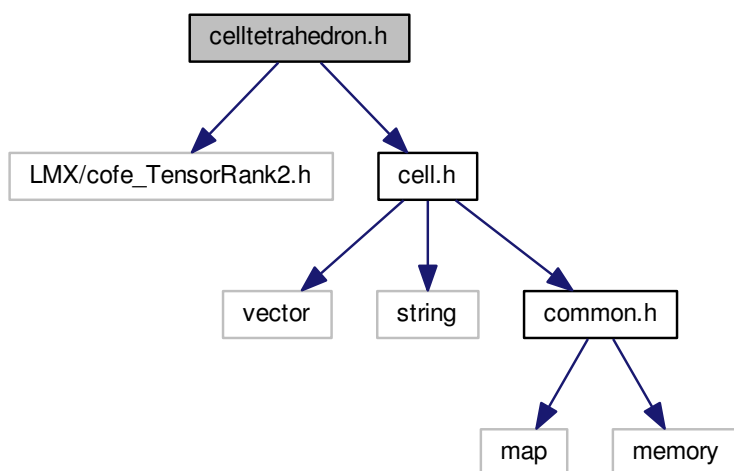
- [mknix](#)

## 7.44 celltetrahedron.h File Reference

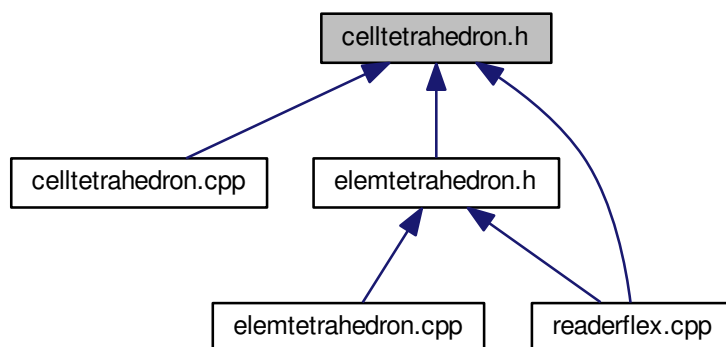
Background cells for integration.

```
#include "LMX/cofe_TensorRank2.h"
#include "cell.h"
```

Include dependency graph for celltetrahedron.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mnix::CellTetrahedron](#)

## Namespaces

- [mnix](#)

## 7.44.1 Detailed Description

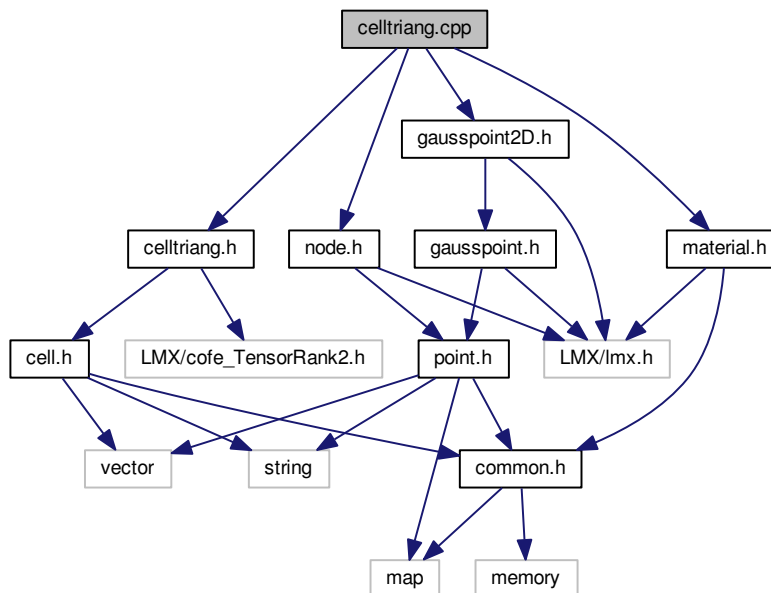
Background cells for integration.

## Author

Daniel Iglesias

## 7.45 celltriang.cpp File Reference

```
#include "celltriang.h"
#include "material.h"
#include "node.h"
#include "gausspoint2D.h"
Include dependency graph for celltriang.cpp:
```



## Namespaces

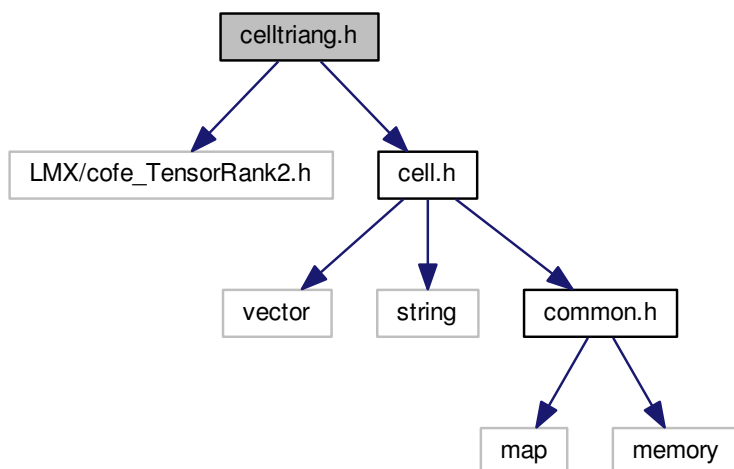
- [mknix](#)

## 7.46 celltriang.h File Reference

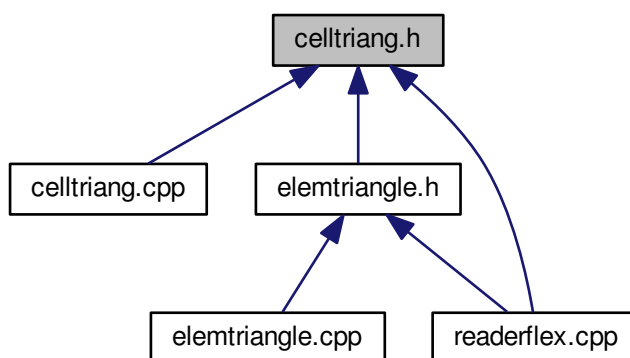
Background cells for integration.

```
#include "LMX/cofe_TensorRank2.h"
#include "cell.h"
```

Include dependency graph for celltriang.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::CellTriang`

#### Namespaces

- `mnix`



## 7.46.1 Detailed Description

Background cells for integration.

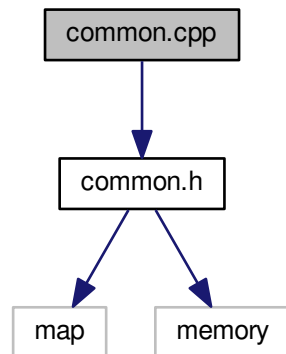
## Author

Daniel Iglesias

## 7.47 common.cpp File Reference

```
#include "common.h"
```

Include dependency graph for common.cpp:



## Namespaces

- [mknix](#)

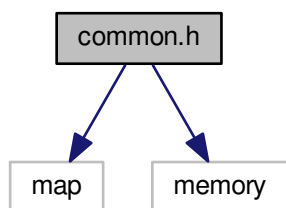
## Functions

- double [mknix::interpolate1D](#) (double key, const std::map< double, double > &theMap)

## 7.48 common.h File Reference

```
#include <map>
#include <memory>
```

Include dependency graph for common.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [mknix](#)

## Typedefs

- typedef double [mknix::data\\_type](#)

## Functions

- double [mknix::interpolate1D](#) (double key, const std::map< double, double > &theMap)
- template<class T , class... Args>  
std::unique\_ptr< T > [mknix::make\\_unique](#) (Args &&...args)

## 7.49 compbar.cpp File Reference

## 7.50 compbar.h File Reference

## Classes

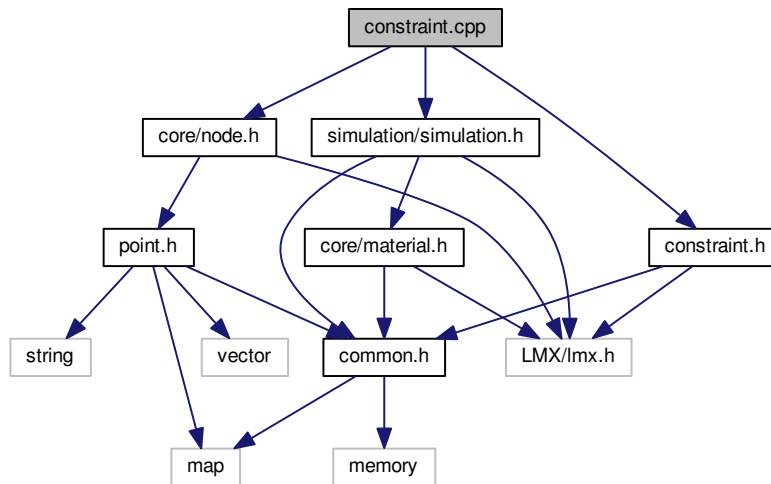
- class [mknix::CompBar](#)

## Namespaces

- [mknix](#)

## 7.51 constraint.cpp File Reference

```
#include "constraint.h"  
#include <core/node.h>  
#include <simulation/simulation.h>  
Include dependency graph for constraint.cpp:
```



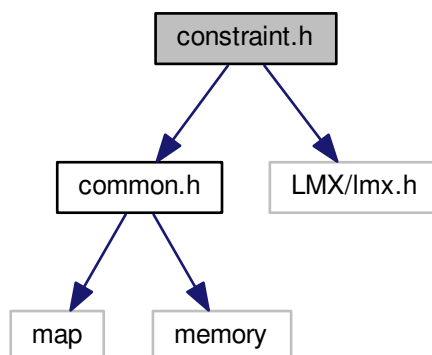
## Namespaces

- [mknix](#)

## 7.52 constraint.h File Reference

```
#include "common.h"  
#include "LMX/lmx.h"
```

Include dependency graph for `constraint.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `mknix::Constraint`

## Namespaces

- `mknix`

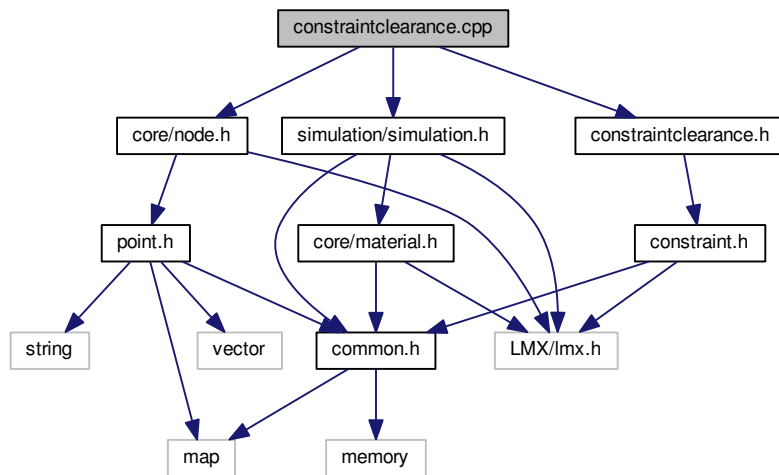
## 7.53 constraintclearance.cpp File Reference

```

#include "constraintclearance.h"
#include <core/node.h>
#include <simulation/simulation.h>

```

Include dependency graph for constraintclearance.cpp:



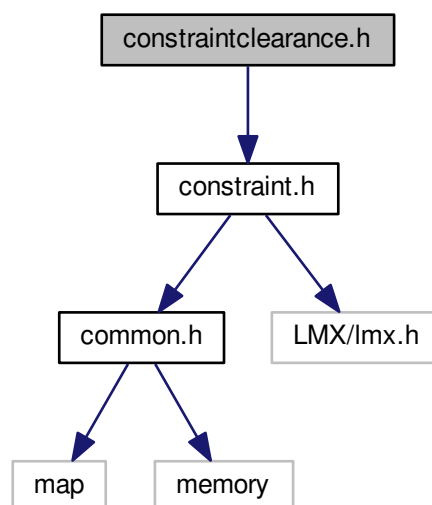
#### Namespaces

- [mknix](#)

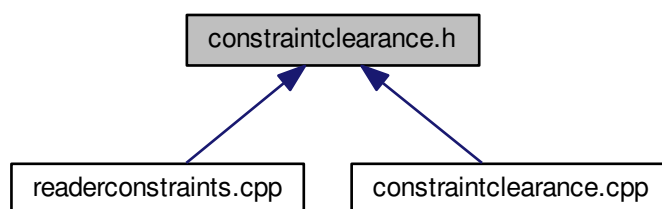
#### 7.54 constraintclearance.h File Reference

```
#include "constraint.h"
```

Include dependency graph for constraintclearance.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::ConstraintClearance`

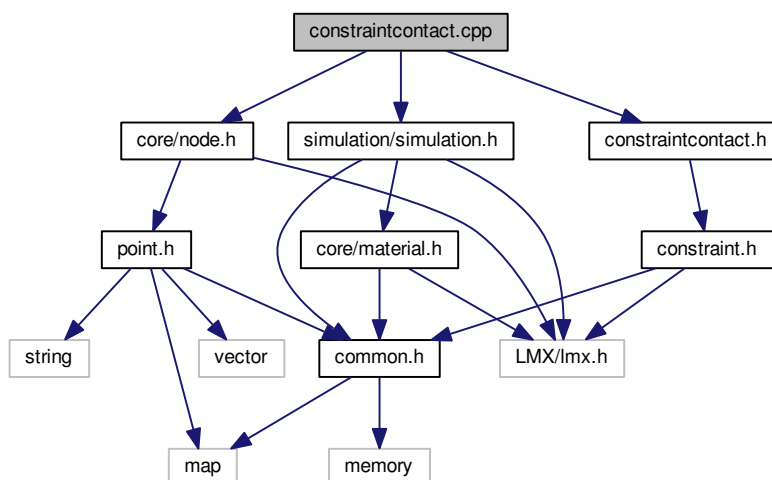
#### Namespaces

- `mnix`

### 7.55 constraintcontact.cpp File Reference

```
#include "constraintcontact.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for `constraintcontact.cpp`:



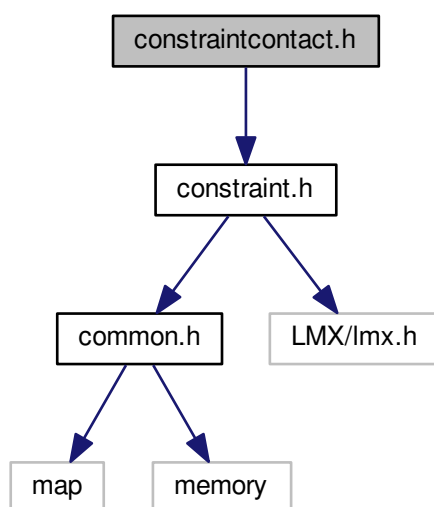
## Namespaces

- [mknix](#)

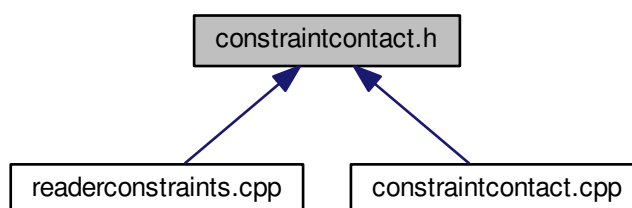
## 7.56 constraintcontact.h File Reference

```
#include "constraint.h"
```

Include dependency graph for constraintcontact.h:



This graph shows which files directly or indirectly include this file:



## Classes

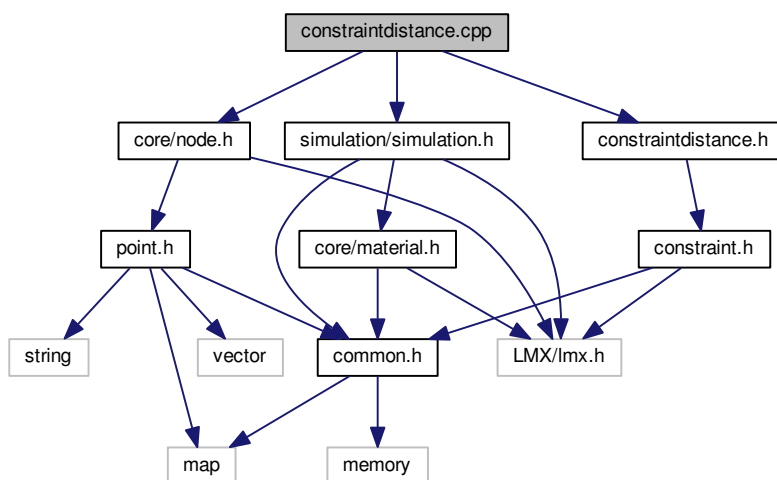
- class [mknix::ConstraintContact](#)

## Namespaces

- [mknix](#)

## 7.57 constraintdistance.cpp File Reference

```
#include "constraintdistance.h"
#include <core/node.h>
#include <simulation/simulation.h>
Include dependency graph for constraintdistance.cpp:
```



## Namespaces

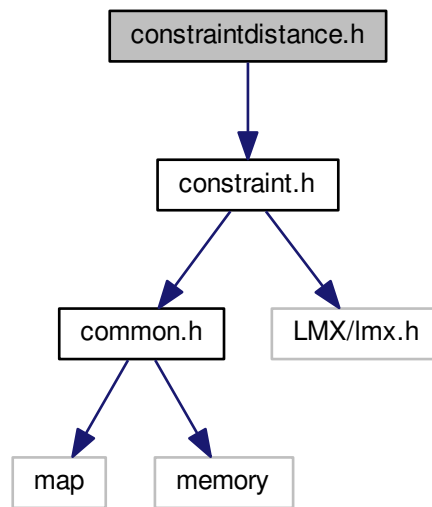
- [mknix](#)

## 7.58 constraintdistance.h File Reference

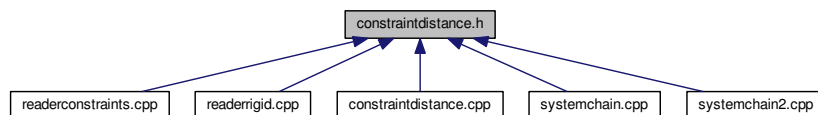
```
#include "constraint.h"
```



Include dependency graph for constraintdistance.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::ConstraintDistance](#)

#### Namespaces

- [mknix](#)

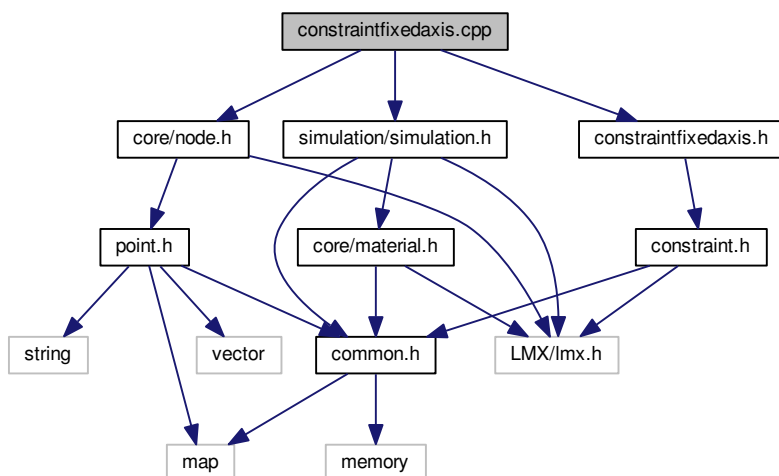
## 7.59 constraintfixedaxis.cpp File Reference

```

#include "constraintfixedaxis.h"
#include <core/node.h>
#include <simulation/simulation.h>

```

Include dependency graph for constraintfixedaxis.cpp:



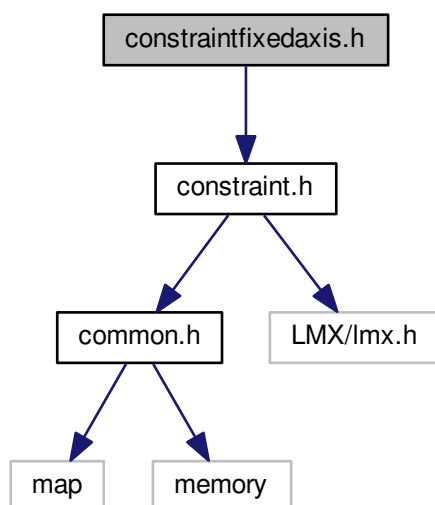
## Namespaces

- [mknix](#)

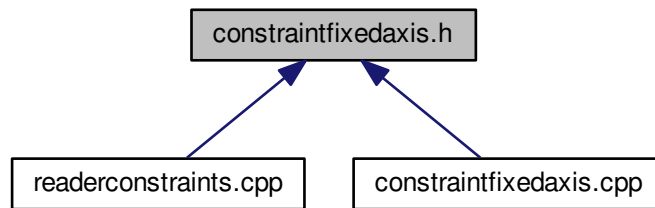
## 7.60 constraintfixedaxis.h File Reference

```
#include "constraint.h"
```

Include dependency graph for constraintfixedaxis.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::ConstraintFixedAxis`

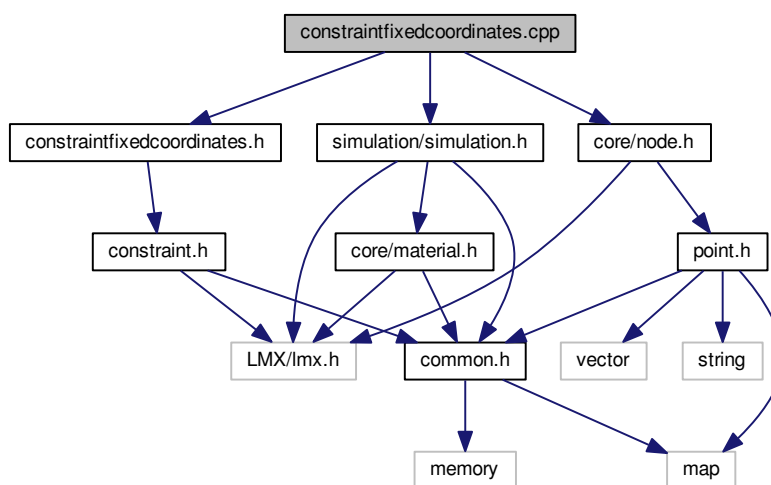
#### Namespaces

- `mnix`

## 7.61 constraintfixedcoordinates.cpp File Reference

```
#include "constraintfixedcoordinates.h"  
#include <simulation/simulation.h>  
#include <core/node.h>
```

Include dependency graph for `constraintfixedcoordinates.cpp`:



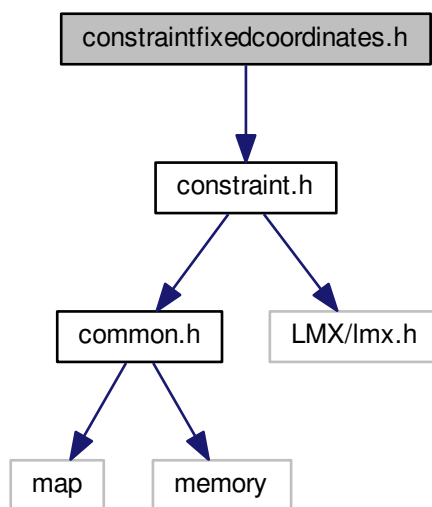
## Namespaces

- [mknix](#)

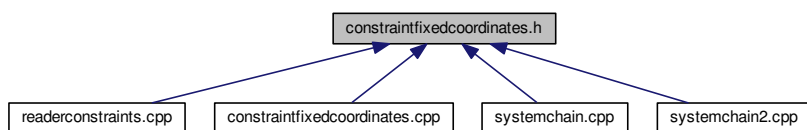
## 7.62 constraintfixedcoordinates.h File Reference

```
#include "constraint.h"
```

Include dependency graph for constraintfixedcoordinates.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mknix::ConstraintFixedCoordinates](#)

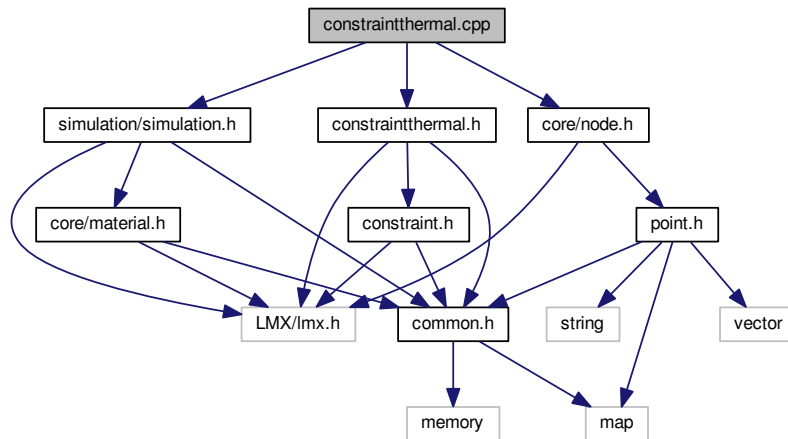
## Namespaces

- [mknix](#)

## 7.63 constraintthermal.cpp File Reference

```
#include "constraintthermal.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for constraintthermal.cpp:



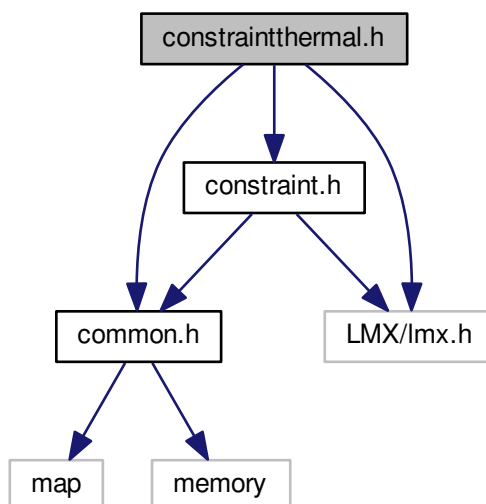
## Namespaces

- [mknix](#)

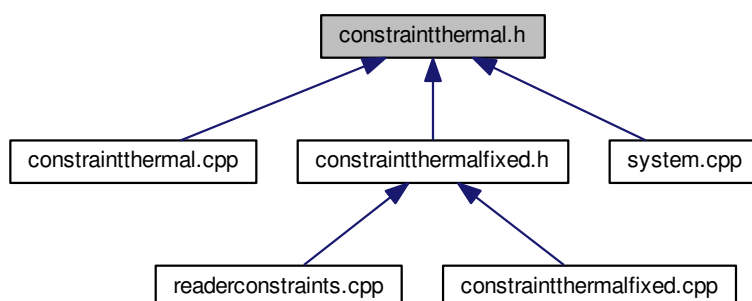
## 7.64 constraintthermal.h File Reference

```
#include "common.h"
#include "LMX/lmx.h"
#include "constraint.h"
```

Include dependency graph for `constraintthermal.h`:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::ConstraintThermal`

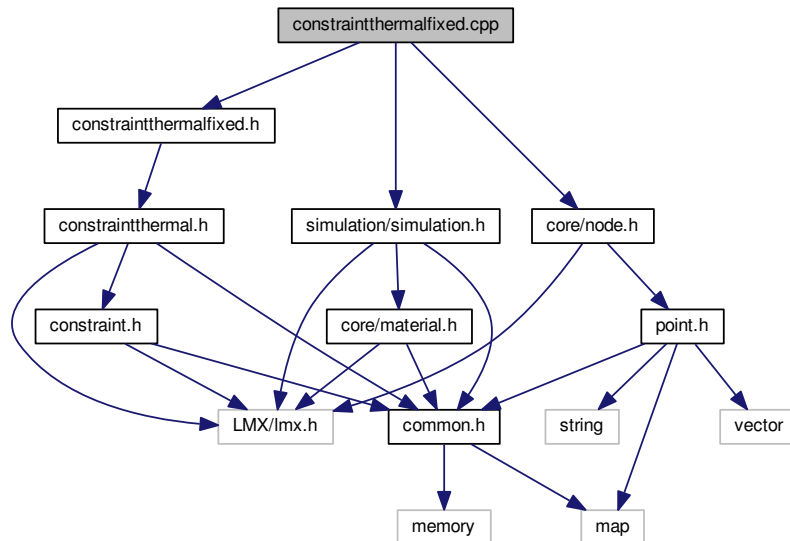
#### Namespaces

- `mnix`

### 7.65 `constraintthermalfixed.cpp` File Reference

```
#include "constraintthermalfixed.h"
```

```
#include <core/node.h>
#include <simulation/simulation.h>
Include dependency graph for constraintthermalfixed.cpp:
```



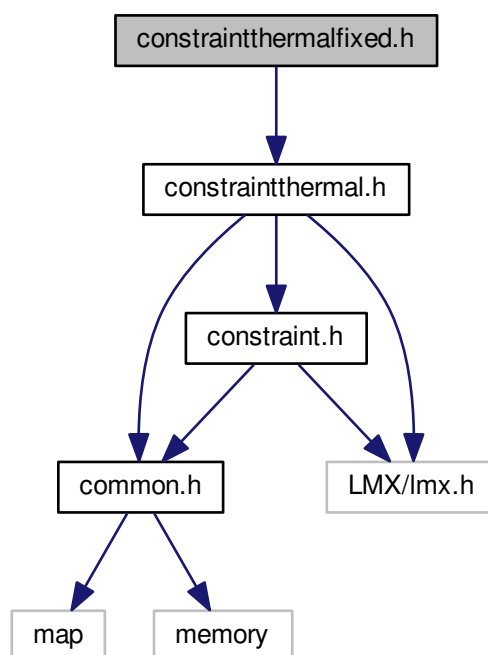
## Namespaces

- [mknix](#)

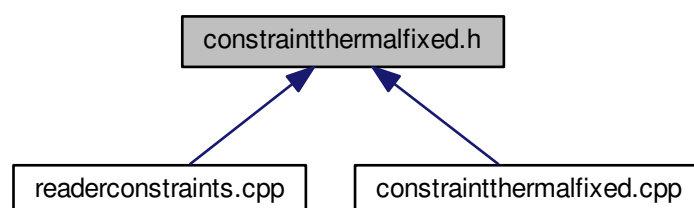
## 7.66 constraintthermalfixed.h File Reference

```
#include "constraintthermal.h"
```

Include dependency graph for `constraintthermalfixed.h`:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mknix::ConstraintThermalFixed`

#### Namespaces

- `mknix`



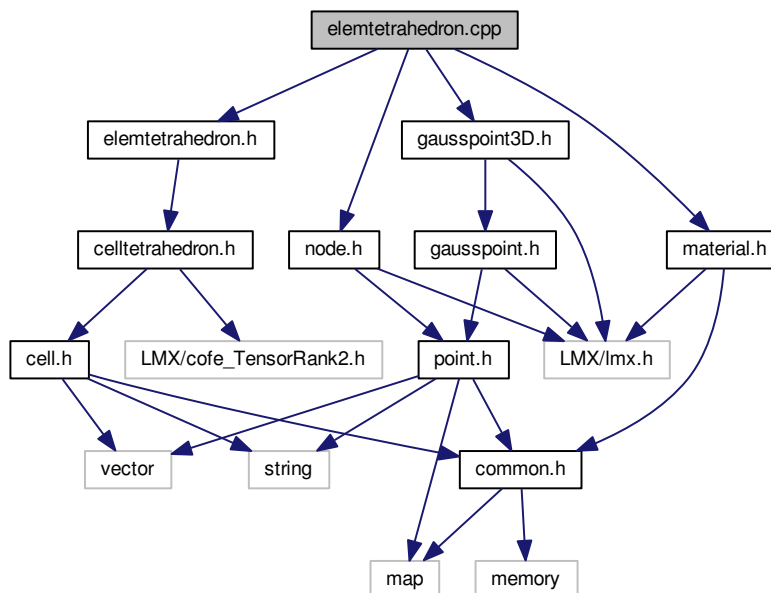
## 7.67 dictionary.h File Reference

## Namespaces

- [mknix](#)

## 7.68 elemtetrahedron.cpp File Reference

```
#include "elemtetrahedron.h"
#include "material.h"
#include "node.h"
#include "gausspoint3D.h"
Include dependency graph for elemtetrahedron.cpp:
```



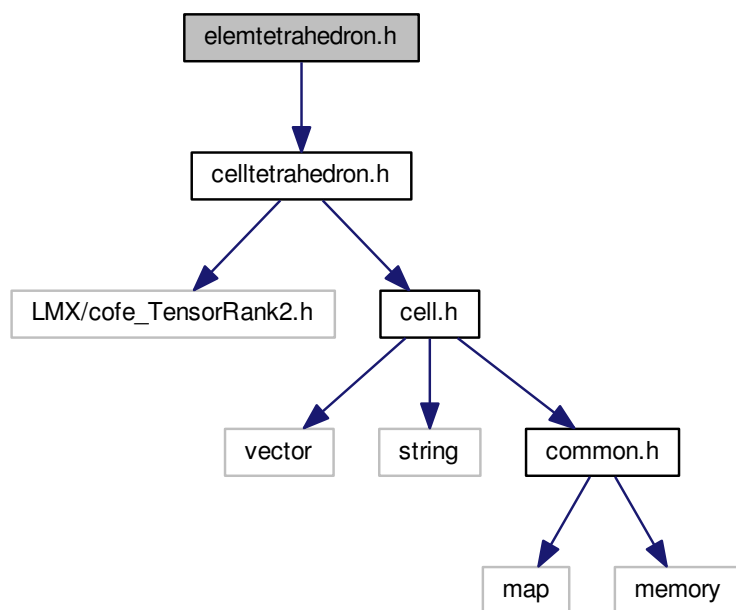
## Namespaces

- [mknix](#)

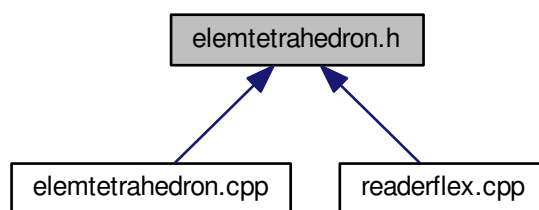
## 7.69 elemtetrahedron.h File Reference

```
#include "celltetrahedron.h"
```

Include dependency graph for `elemtetrahedron.h`:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::ElemTetrahedron`

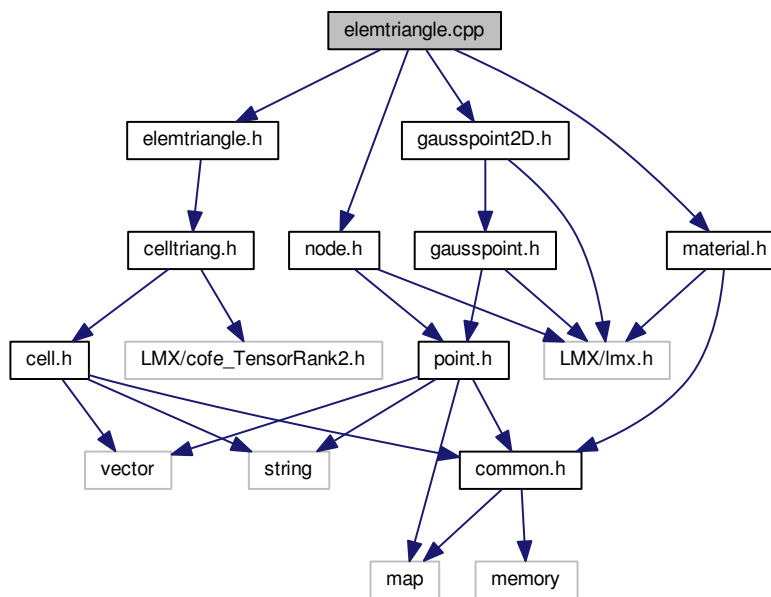
#### Namespaces

- `mnix`

## 7.70 elemtriangle.cpp File Reference

```
#include "elemtriangle.h"  
#include "material.h"  
#include "node.h"  
#include "gausspoint2D.h"
```

Include dependency graph for elemtriangle.cpp:



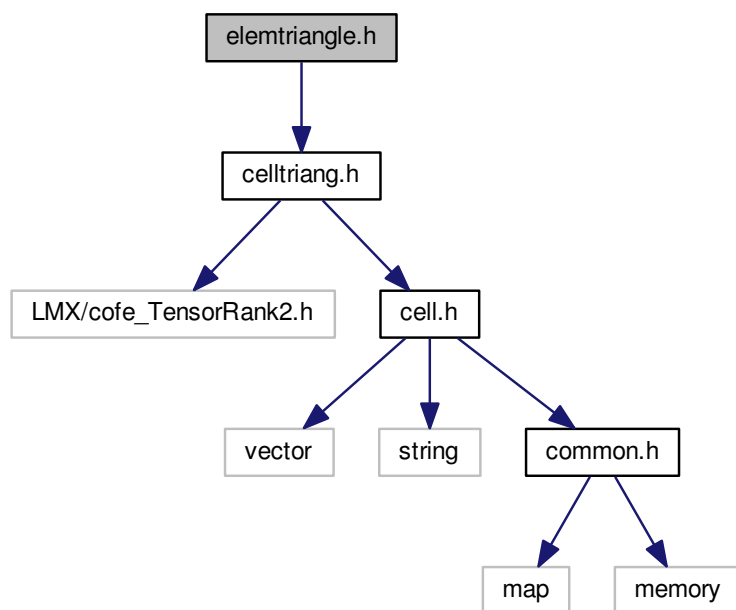
## Namespaces

- [mknix](#)

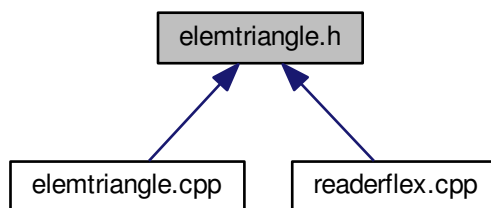
## 7.71 elemtriangle.h File Reference

```
#include "celltriang.h"
```

Include dependency graph for elemtriangle.h:



This graph shows which files directly or indirectly include this file:



#### Classes

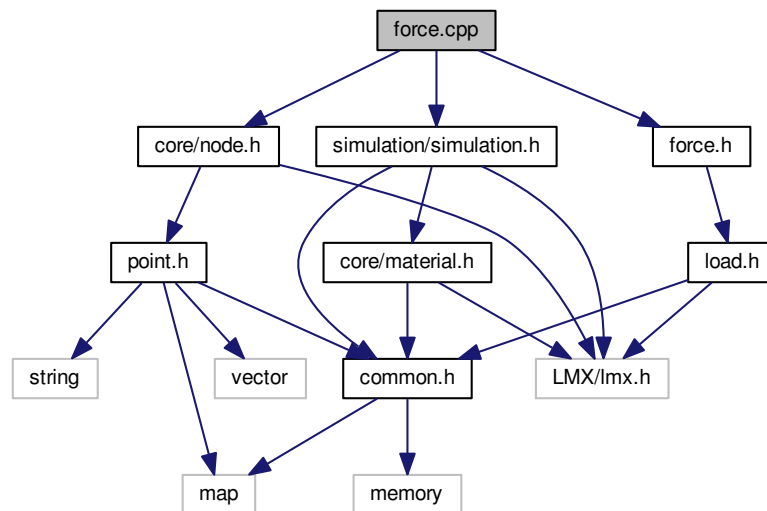
- class `mknix::ElemTriangle`

#### Namespaces

- `mknix`

## 7.72 force.cpp File Reference

```
#include "force.h"  
#include <core/node.h>  
#include <simulation/simulation.h>  
Include dependency graph for force.cpp:
```



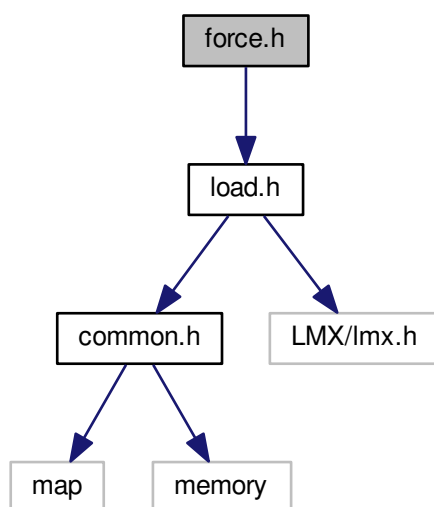
## Namespaces

- [mknix](#)

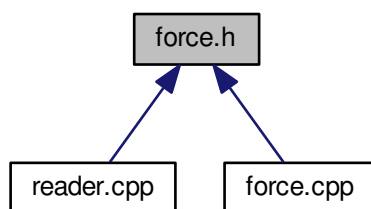
## 7.73 force.h File Reference

```
#include "load.h"
```

Include dependency graph for force.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mnix::Force](#)

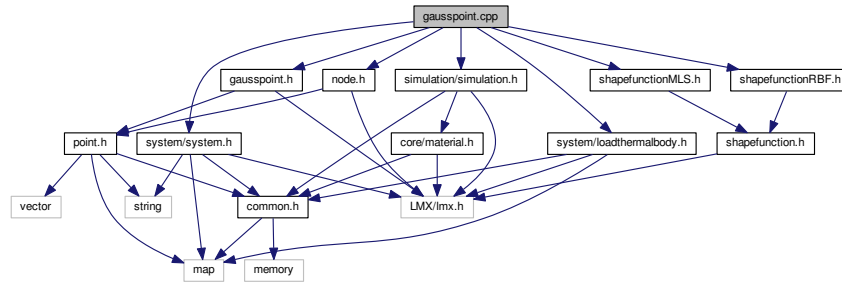
#### Namespaces

- [mnix](#)

### 7.74 gausspoint.cpp File Reference

```
#include "gausspoint.h"
```

```
#include "node.h"
#include "shapefunctionRBF.h"
#include "shapefunctionMLS.h"
#include <simulation/simulation.h>
#include <system/system.h>
#include <system/loadthermalbody.h>
Include dependency graph for gausspoint.cpp:
```



## Namespaces

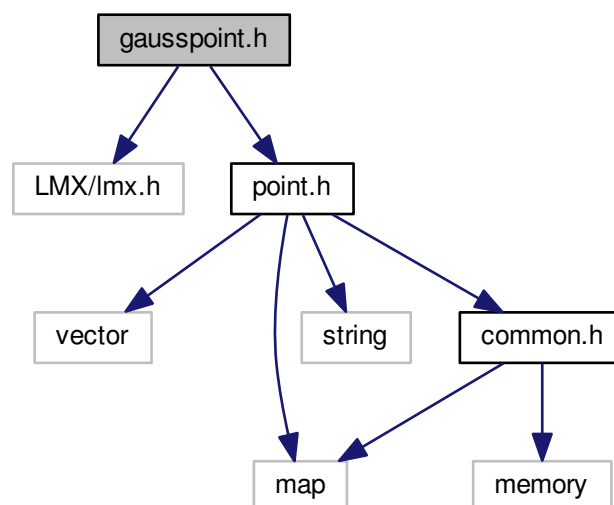
- [mknix](#)

## 7.75 gausspoint.h File Reference

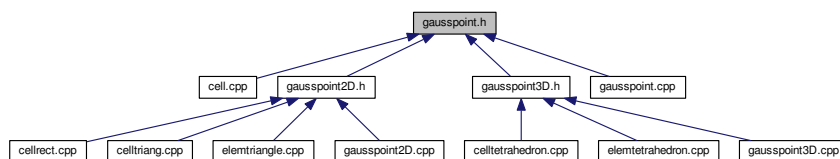
Point for numerical integration.

```
#include "LMX/lmx.h"
#include "point.h"
```

Include dependency graph for gausspoint.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mknix::GaussPoint](#)

## Namespaces

- [mknix](#)

### 7.75.1 Detailed Description

Point for numerical integration.

## Author

Daniel Iglesias

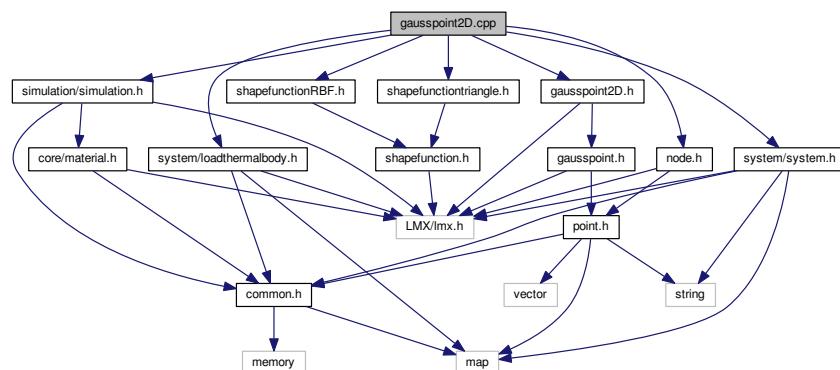
### 7.76 gausspoint2D.cpp File Reference

```

#include "gausspoint2D.h"
#include "node.h"
#include "shapefunctionRBF.h"
#include "shapefunctiontriangle.h"
#include <simulation/simulation.h>
#include <system/system.h>
#include <system/loadthermalbody.h>

```

Include dependency graph for gausspoint2D.cpp:





## Namespaces

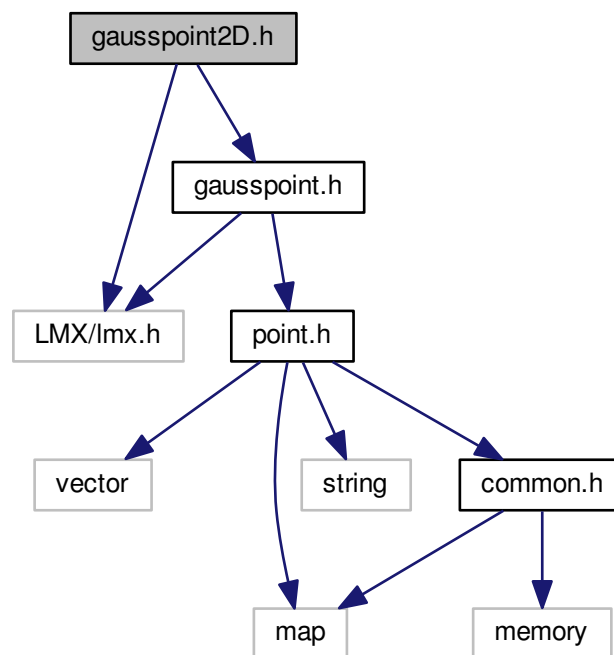
- [mknix](#)

## 7.77 gausspoint2D.h File Reference

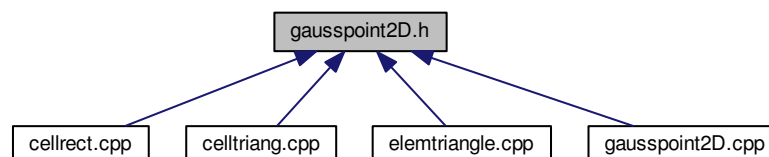
```
#include "LMX/lmx.h"
```

```
#include "gausspoint.h"
```

Include dependency graph for gausspoint2D.h:



This graph shows which files directly or indirectly include this file:



## Classes

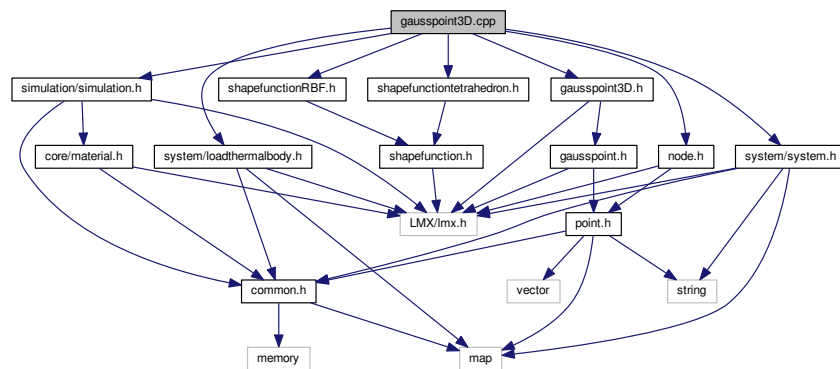
- class [mknix::GaussPoint2D](#)

## Namespaces

- [mknix](#)

## 7.78 gausspoint3D.cpp File Reference

```
#include "gausspoint3D.h"
#include "node.h"
#include "shapefunctionRBF.h"
#include "shapefunctiontetrahedron.h"
#include <simulation/simulation.h>
#include <system/system.h>
#include <system/loadthermalbody.h>
Include dependency graph for gausspoint3D.cpp:
```



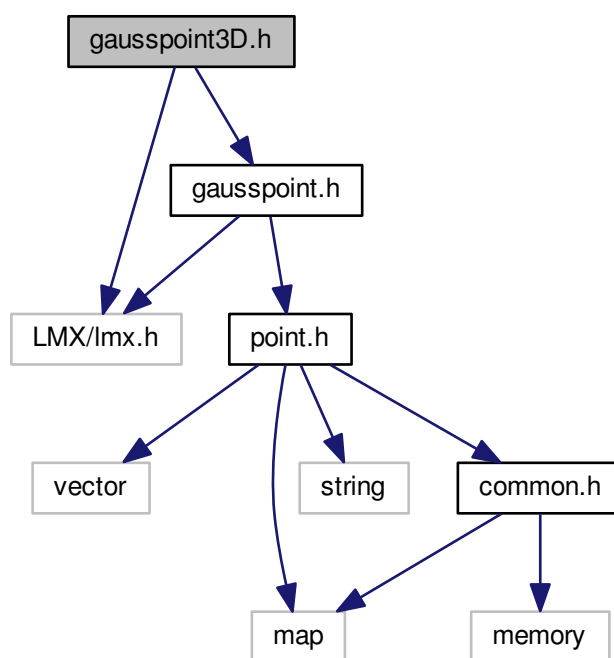
## Namespaces

- [mknix](#)

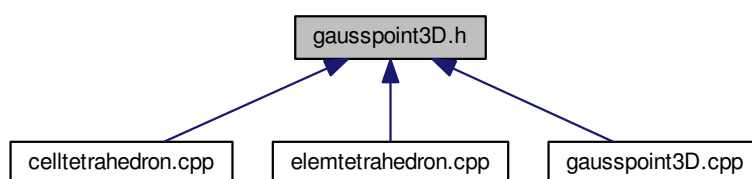
## 7.79 gausspoint3D.h File Reference

```
#include "LMX/lmx.h"
#include "gausspoint.h"
```

Include dependency graph for gausspoint3D.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mnix::GaussPoint3D](#)

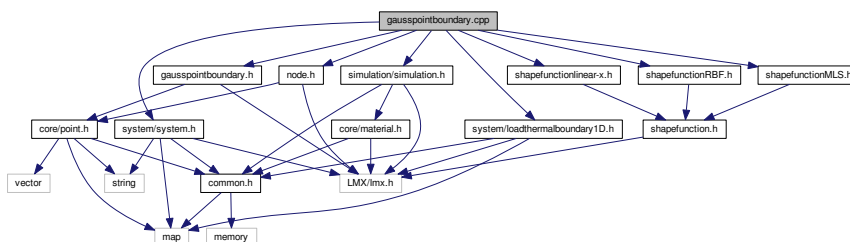
#### Namespaces

- [mnix](#)

## 7.80 gausspointboundary.cpp File Reference

```
#include "gausspointboundary.h"
#include "node.h"
#include "shapefunctionRBF.h"
#include "shapefunctionMLS.h"
#include "shapefunctionlinear-x.h"
#include <simulation/simulation.h>
#include <system/system.h>
#include <system/loadthermalboundary1D.h>
```

Include dependency graph for gausspointboundary.cpp:



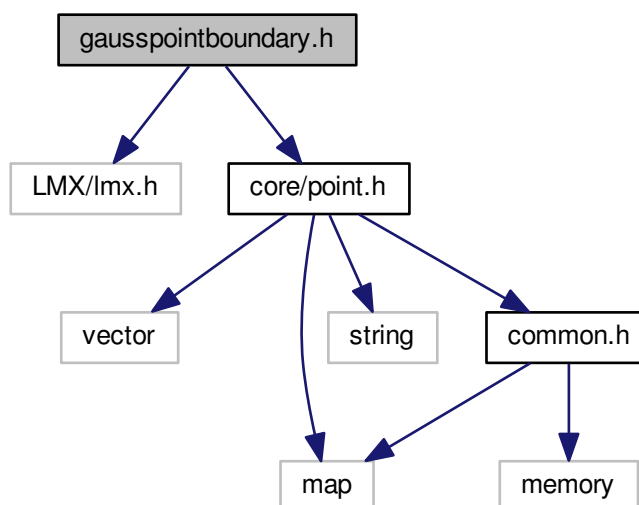
### Namespaces

- [mknix](#)

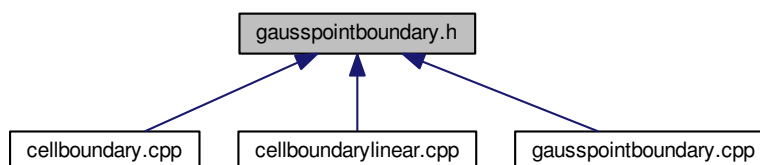
## 7.81 gausspointboundary.h File Reference

```
#include <LMX/lmx.h>
#include <core/point.h>
```

Include dependency graph for gausspointboundary.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mknix::GaussPointBoundary`

#### Namespaces

- `mknix`

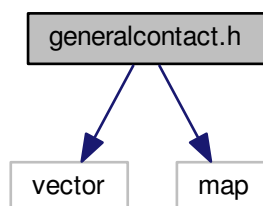
## 7.82 `generalcontact.cpp` File Reference

## 7.83 `generalcontact.h` File Reference

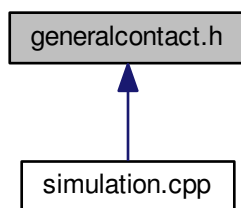
```
#include <vector>
```

```
#include <map>
```

Include dependency graph for `generalcontact.h`:



This graph shows which files directly or indirectly include this file:



#### Classes

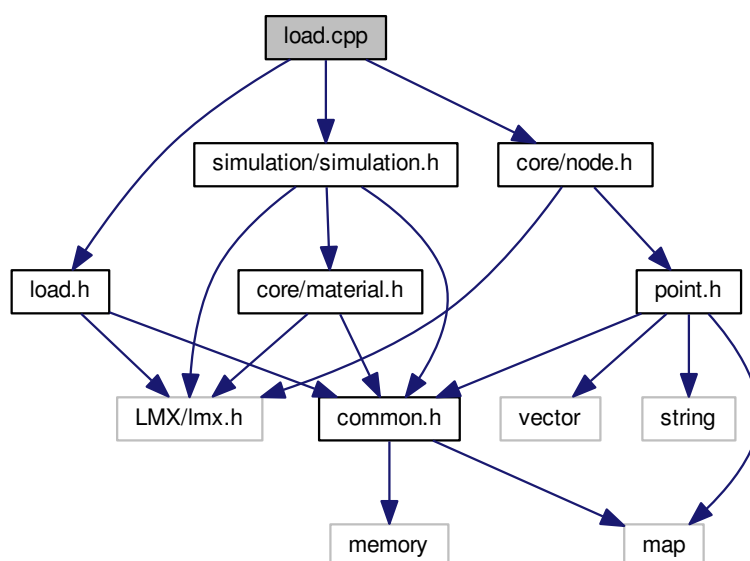
- class `mnix::Contact`

#### Namespaces

- `mnix`

### 7.84 load.cpp File Reference

```
#include "load.h"
#include <simulation/simulation.h>
#include <core/node.h>
Include dependency graph for load.cpp:
```



## Namespaces

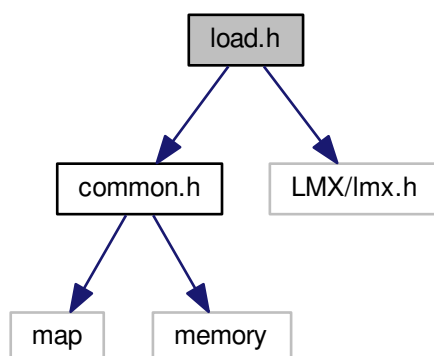
- [mknix](#)

## 7.85 load.h File Reference

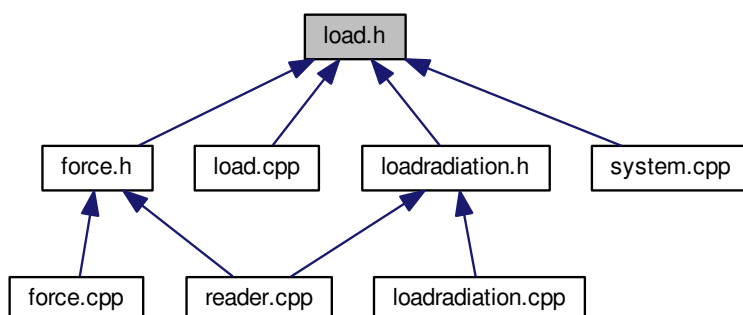
```
#include "common.h"
```

```
#include "LMX/lmx.h"
```

Include dependency graph for load.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mknix::Load](#)

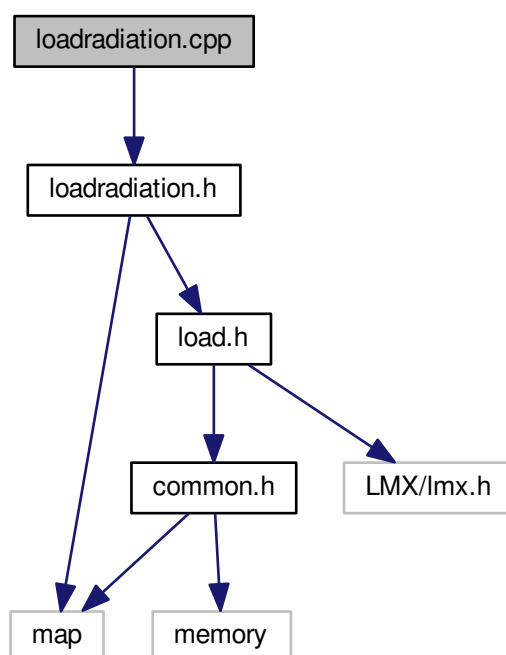
## Namespaces

- [mknix](#)

### 7.86 loadradiation.cpp File Reference

```
#include "loadradiation.h"
```

Include dependency graph for loadradiation.cpp:



#### Namespaces

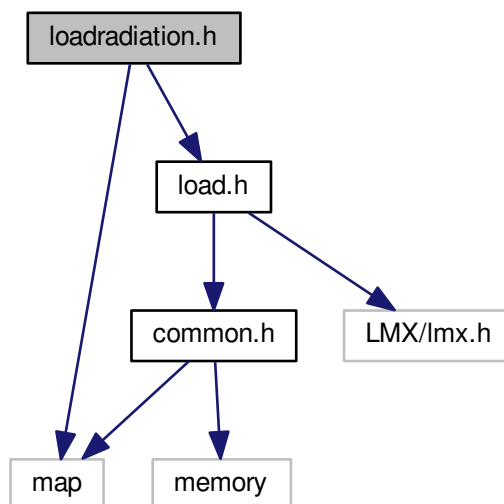
- [mknix](#)

### 7.87 loadradiation.h File Reference

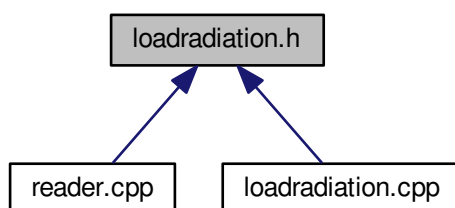
```
#include <map>
#include "load.h"
```



Include dependency graph for loadradiation.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::Radiation](#)

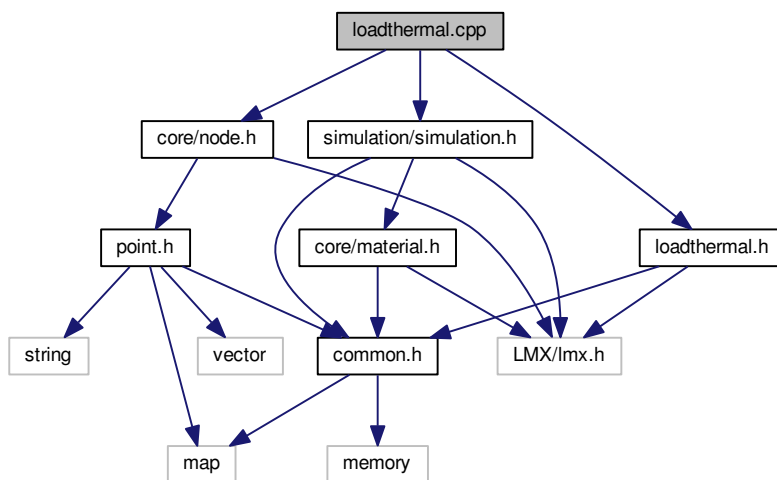
#### Namespaces

- [mknix](#)

## 7.88 loadthermal.cpp File Reference

```
#include "loadthermal.h"  
#include <core/node.h>  
#include <simulation/simulation.h>
```

Include dependency graph for loadthermal.cpp:



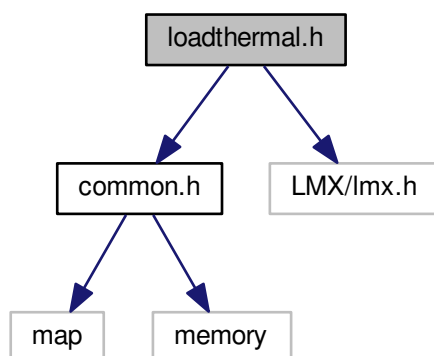
## Namespaces

- [mknix](#)

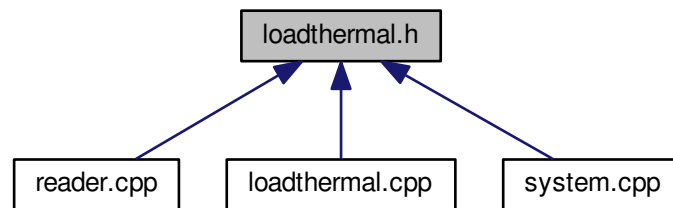
## 7.89 loadthermal.h File Reference

```
#include "common.h"
#include "LMX/lmx.h"
```

Include dependency graph for loadthermal.h:



This graph shows which files directly or indirectly include this file:



#### Classes

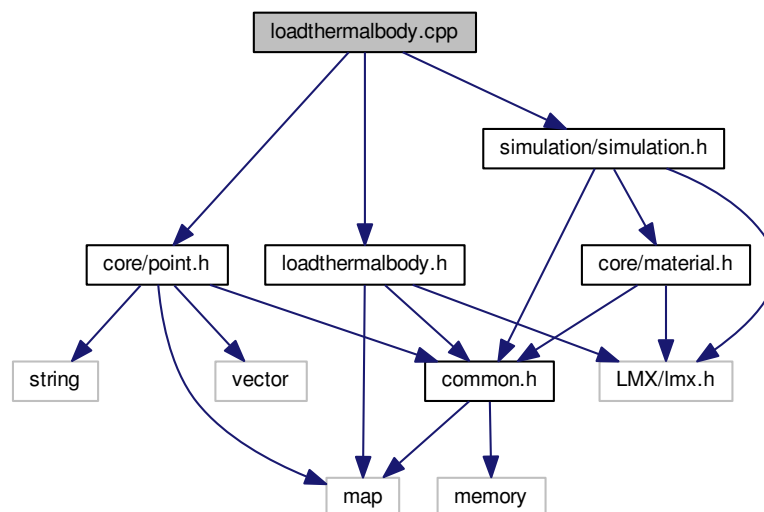
- class [mknix::LoadThermal](#)

#### Namespaces

- [mknix](#)

### 7.90 loadthermalbody.cpp File Reference

```
#include "loadthermalbody.h"  
#include <core/point.h>  
#include <simulation/simulation.h>  
Include dependency graph for loadthermalbody.cpp:
```



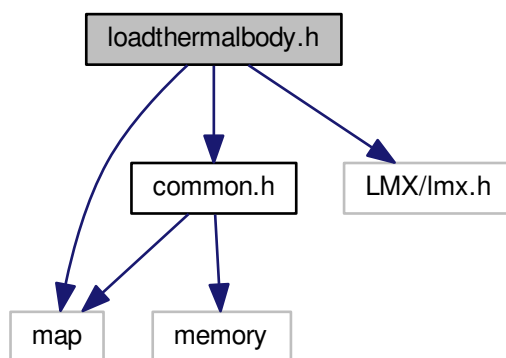
## Namespaces

- [mknix](#)

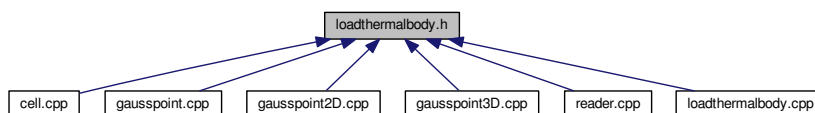
## 7.91 loadthermalbody.h File Reference

```
#include "common.h"
#include "LMX/lmx.h"
#include <map>
```

Include dependency graph for loadthermalbody.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mknix::LoadThermalBody](#)

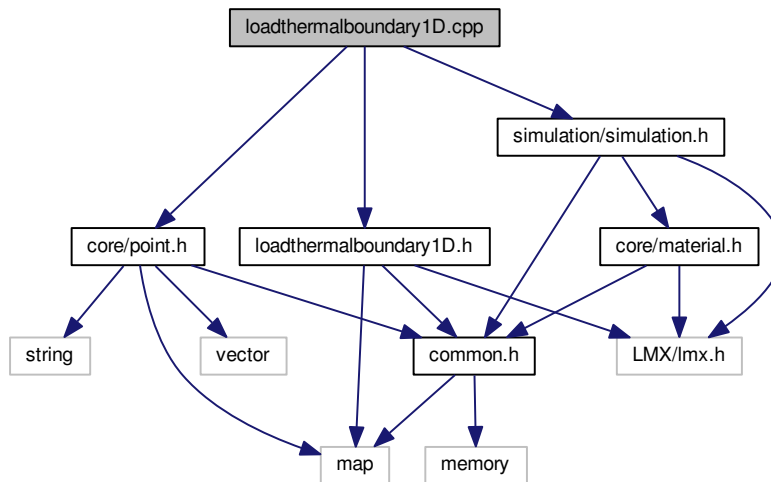
## Namespaces

- [mknix](#)

## 7.92 loadthermalboundary1D.cpp File Reference

```
#include "loadthermalboundary1D.h"
#include <core/point.h>
#include <simulation/simulation.h>
```

Include dependency graph for loadthermalboundary1D.cpp:



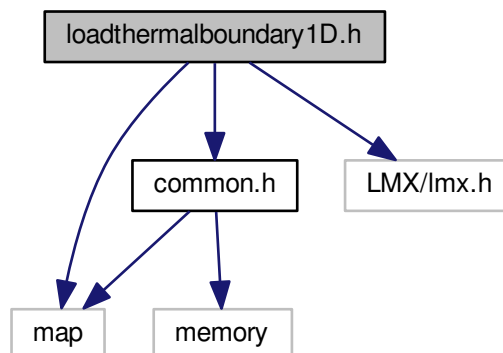
#### Namespaces

- [mknix](#)

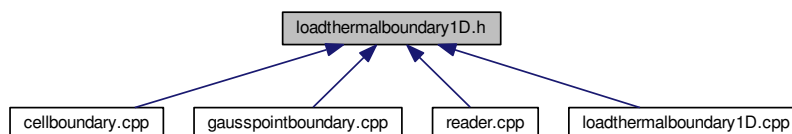
#### 7.93 loadthermalboundary1D.h File Reference

```
#include "common.h"
#include "LMX/lmx.h"
#include <map>
```

Include dependency graph for loadthermalboundary1D.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::LoadThermalBoundary1D](#)

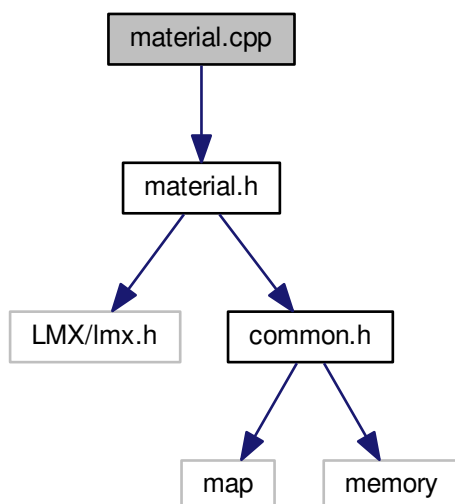
#### Namespaces

- [mknix](#)

### 7.94 material.cpp File Reference

```
#include "material.h"
```

Include dependency graph for material.cpp:

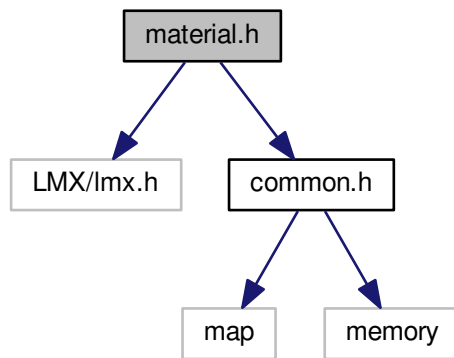


#### Namespaces

- [mknix](#)

## 7.95 material.h File Reference

```
#include "LMX/lmx.h"  
#include "common.h"  
Include dependency graph for material.h:
```



This graph shows which files directly or indirectly include this file:



### Classes

- class [mknix::Material](#)

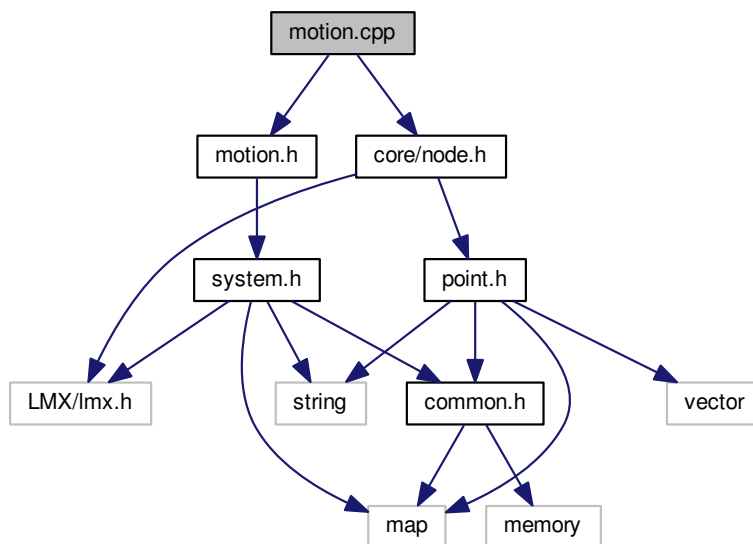
### Namespaces

- [mknix](#)

## 7.96 motion.cpp File Reference

```
#include "motion.h"  
#include <core/node.h>
```

Include dependency graph for motion.cpp:



## Namespaces

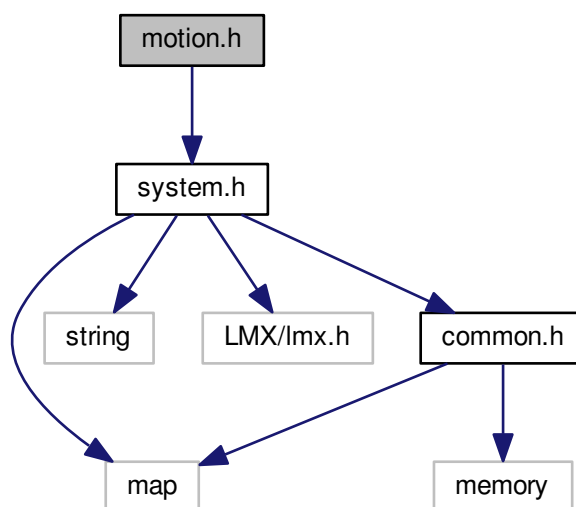
- [mknix](#)

## 7.97 motion.h File Reference

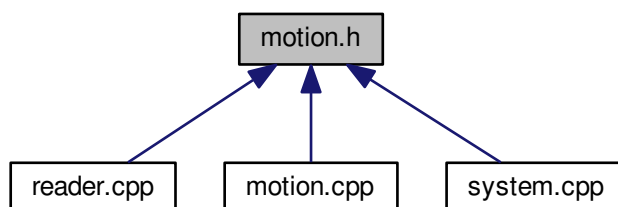
```
#include "system.h"
```



Include dependency graph for motion.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::Motion](#)

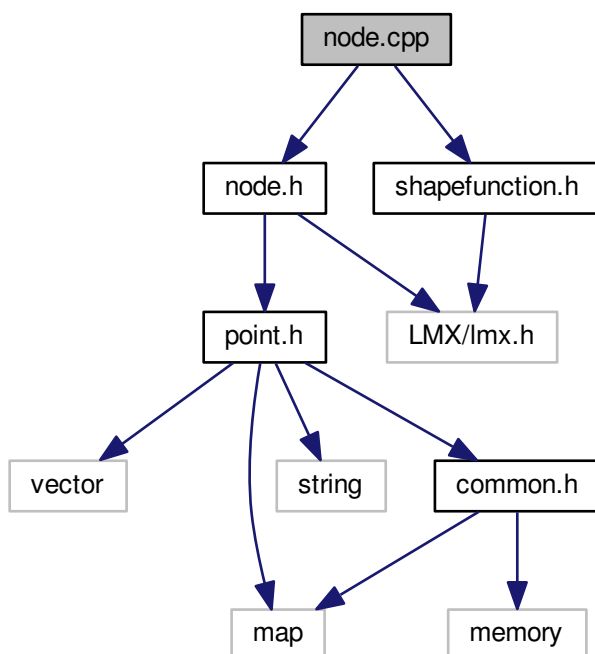
#### Namespaces

- [mknix](#)

## 7.98 node.cpp File Reference

```
#include "node.h"
#include "shapefunction.h"
```

Include dependency graph for node.cpp:



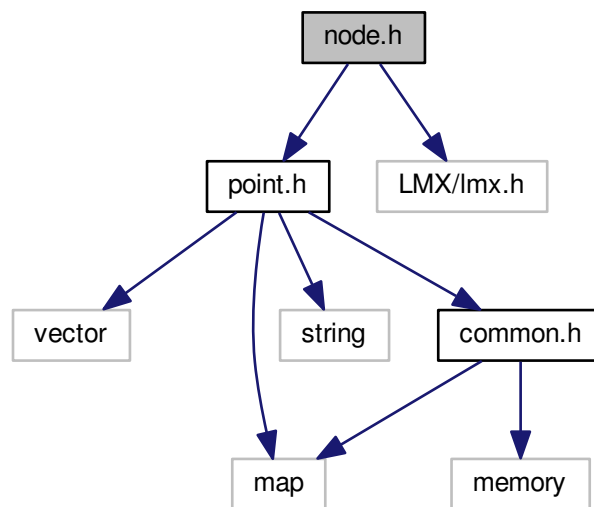
## Namespaces

- [mknix](#)

## 7.99 node.h File Reference

```
#include "point.h"
#include "LMX/lmx.h"
```

Include dependency graph for node.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::Node](#)

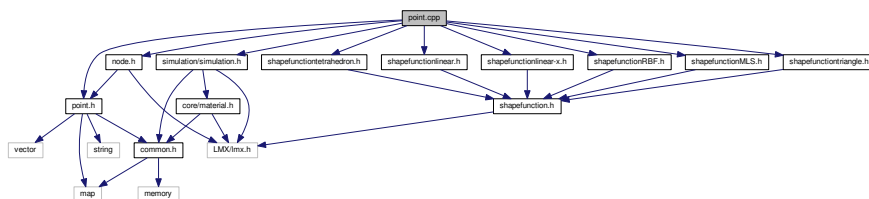
#### Namespaces

- [mknix](#)

## 7.100 point.cpp File Reference

```
#include "point.h"
#include "node.h"
#include "shapefunctionRBF.h"
#include "shapefunctionMLS.h"
#include "shapefunctiontriangle.h"
#include "shapefunctiontetrahedron.h"
#include "shapefunctionlinear.h"
#include "shapefunctionlinear-x.h"
#include <simulation/simulation.h>
```

Include dependency graph for point.cpp:



## Namespaces

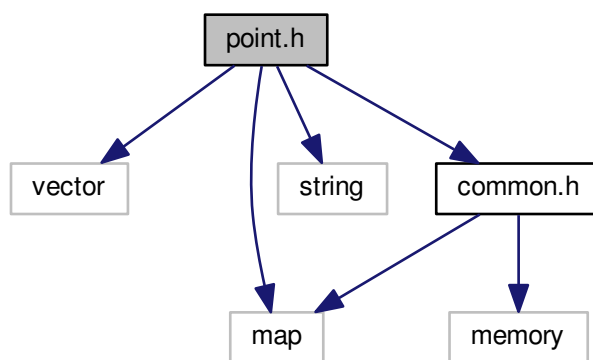
- [mknix](#)

## 7.101 point.h File Reference

Point of interest.

```
#include <vector>
#include <map>
#include <string>
#include "common.h"
```

Include dependency graph for point.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mknix::Point](#)

## Namespaces

- [mknix](#)

## 7.101.1 Detailed Description

Point of interest.

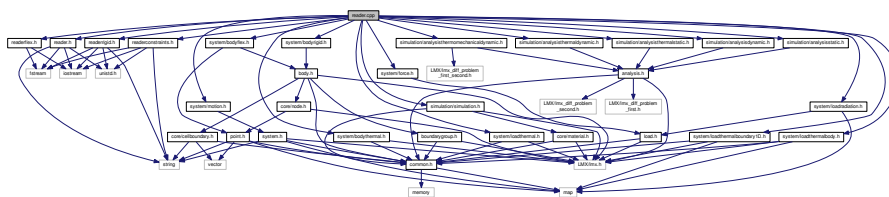
## Author

Daniel Iglesias

## 7.102 reader.cpp File Reference

```
#include "reader.h"
#include "readerrigid.h"
#include "readerflex.h"
#include "readerconstraints.h"
#include <simulation/simulation.h>
#include <simulation/analysisdynamic.h>
#include <simulation/analysisstatic.h>
#include <simulation/analysisthermaldynamic.h>
#include <simulation/analysisthermalstatic.h>
#include <simulation/analysisthermomechanicaldynamic.h>
#include <system/bodyflex.h>
#include <system/bodyrigid.h>
#include <system/bodythermal.h>
#include <system/force.h>
#include <system/loadradiation.h>
#include <system/loadthermal.h>
#include <system/motion.h>
#include <system/loadthermalbody.h>
#include <system/loadthermalboundary1D.h>
```

Include dependency graph for reader.cpp:



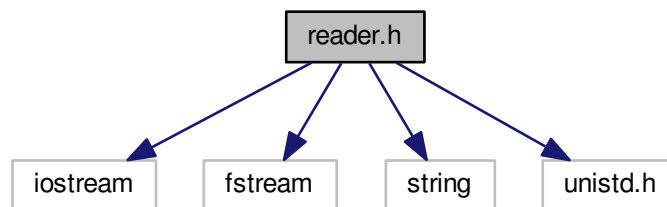
## Namespaces

- [mknix](#)

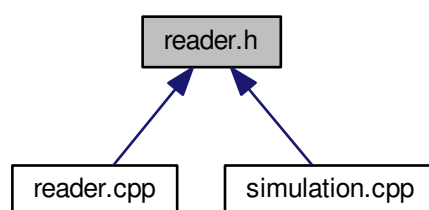
## 7.103 reader.h File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <unistd.h>
```

Include dependency graph for reader.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mnix::Reader](#)

## Namespaces

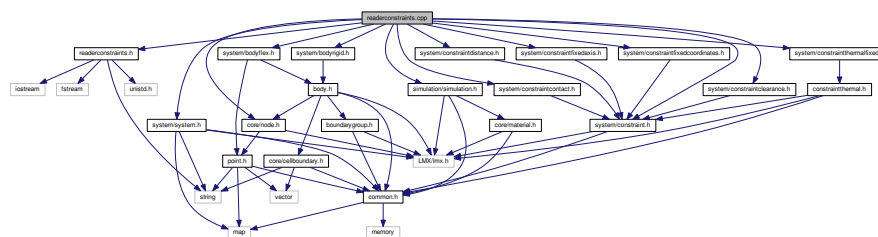
- [mnix](#)

## 7.104 readerconstraints.cpp File Reference

```
#include "readerconstraints.h"
```

```
#include <core/node.h>
#include <simulation/simulation.h>
#include <system/bodyflex.h>
#include <system/bodyrigid.h>
#include <system/constraint.h>
#include <system/constraintclearance.h>
#include <system/constraintcontact.h>
#include <system/constraintdistance.h>
#include <system/constraintfixedaxis.h>
#include <system/constraintfixedcoordinates.h>
#include <system/constraintthermalfixed.h>
#include <system/system.h>
```

Include dependency graph for readerconstraints.cpp:



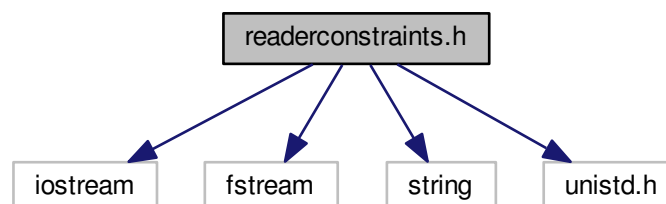
## Namespaces

- `mknix`

## 7.105 readerconstraints.h File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <unistd.h>
```

Include dependency graph for readerconstraints.h:







## Namespaces

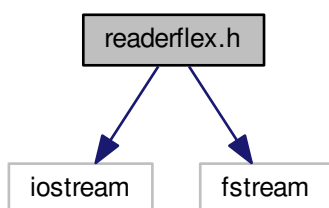
- [mknix](#)

## 7.107 readerflex.h File Reference

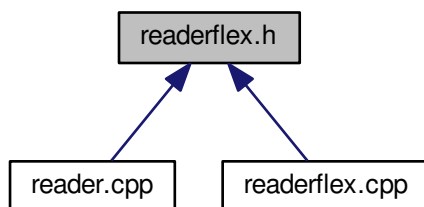
```
#include <iostream>
```

```
#include <fstream>
```

Include dependency graph for readerflex.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mknix::ReaderFlex](#)

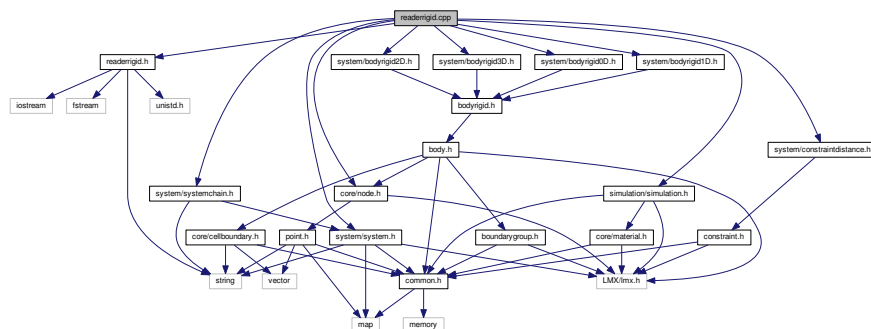
## Namespaces

- [mknix](#)

## 7.108 readerrigid.cpp File Reference

```
#include "readerrigid.h"
```

```
#include <core/node.h>
#include <simulation/simulation.h>
#include <system/bodyrigid0D.h>
#include <system/bodyrigid1D.h>
#include <system/bodyrigid2D.h>
#include <system/bodyrigid3D.h>
#include <system/constraintdistance.h>
#include <system/system.h>
#include <system/systemchain.h>
Include dependency graph for readerrigid.cpp:
```



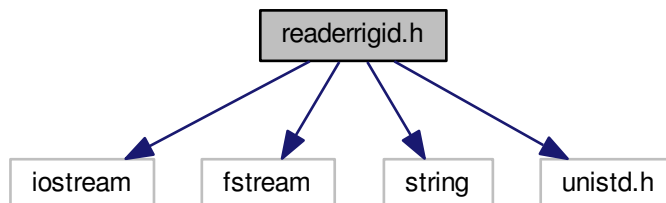
## Namespaces

- [mknix](#)

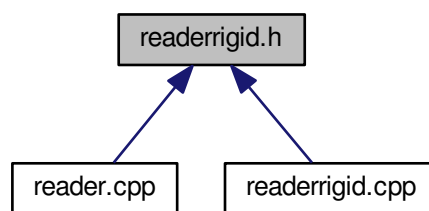
## 7.109 readerrigid.h File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <unistd.h>
```

Include dependency graph for readerrigid.h:



This graph shows which files directly or indirectly include this file:



#### Classes

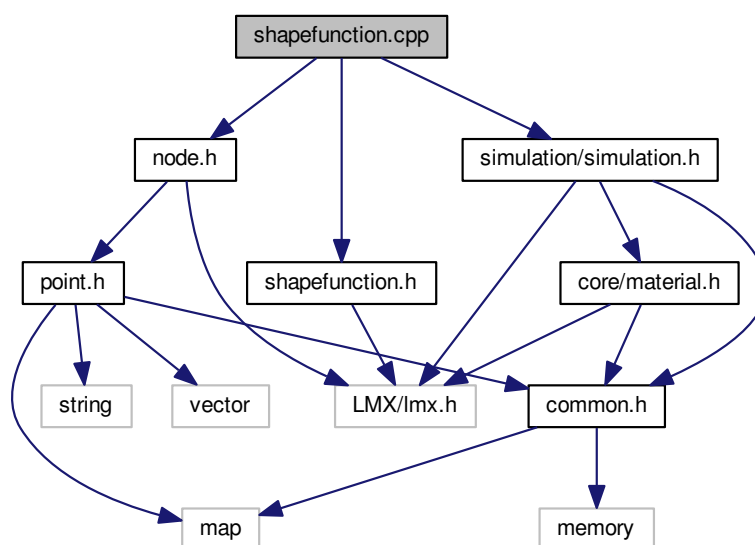
- class `mknix::ReaderRigid`

#### Namespaces

- `mknix`

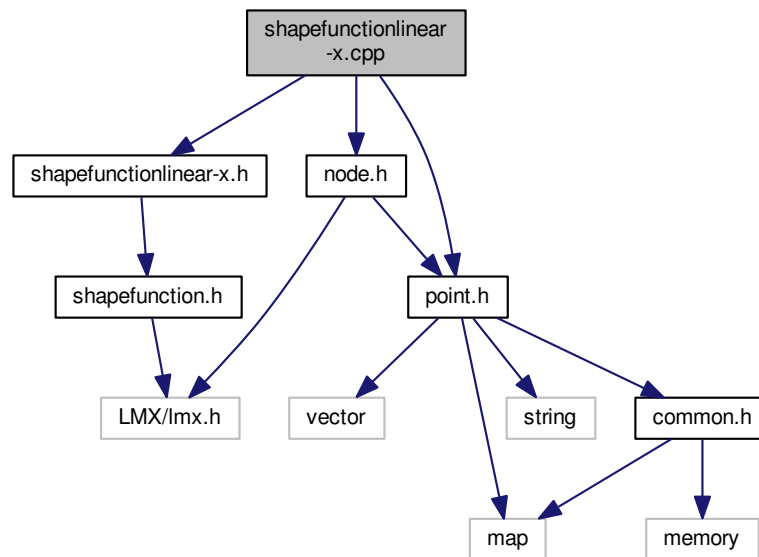
### 7.110 shapefunction.cpp File Reference

```
#include "shapefunction.h"  
#include "node.h"  
#include <simulation/simulation.h>  
Include dependency graph for shapefunction.cpp:
```





Include dependency graph for shapefunctionlinear-x.cpp:



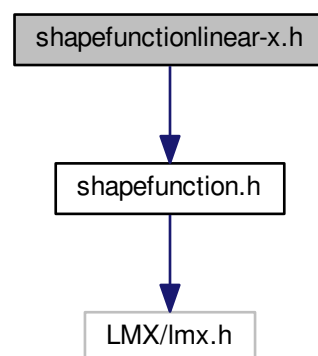
Namespaces

- [mknix](#)

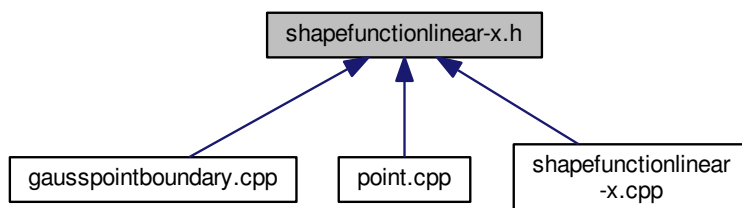
### 7.113 shapefunctionlinear-x.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionlinear-x.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::ShapeFunctionLinearX`

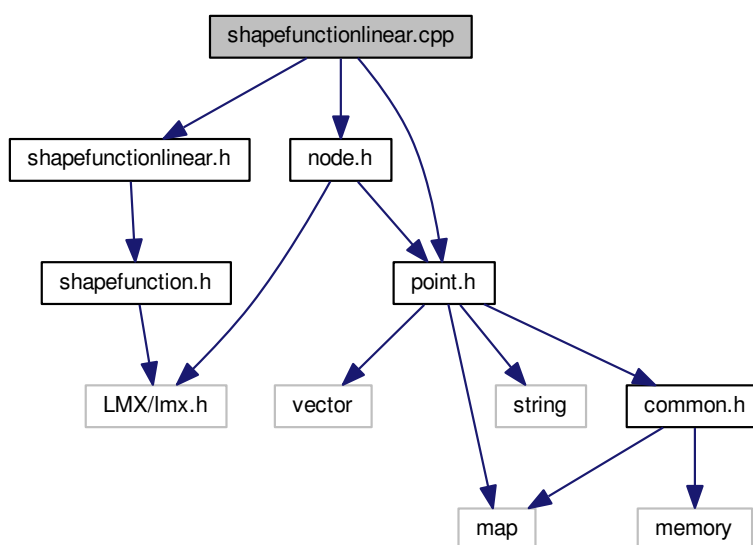
#### Namespaces

- `mnix`

### 7.114 shapefunctionlinear.cpp File Reference

```
#include "shapefunctionlinear.h"  
#include "point.h"  
#include "node.h"
```

Include dependency graph for `shapefunctionlinear.cpp`:



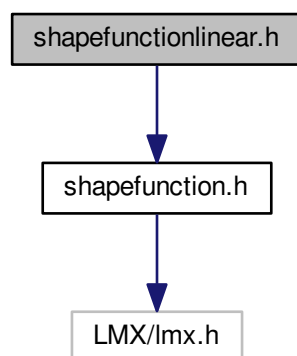
## Namespaces

- [mknix](#)

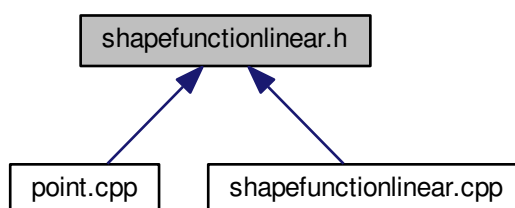
## 7.115 shapefunctionlinear.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionlinear.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mknix::ShapeFunctionLinear](#)

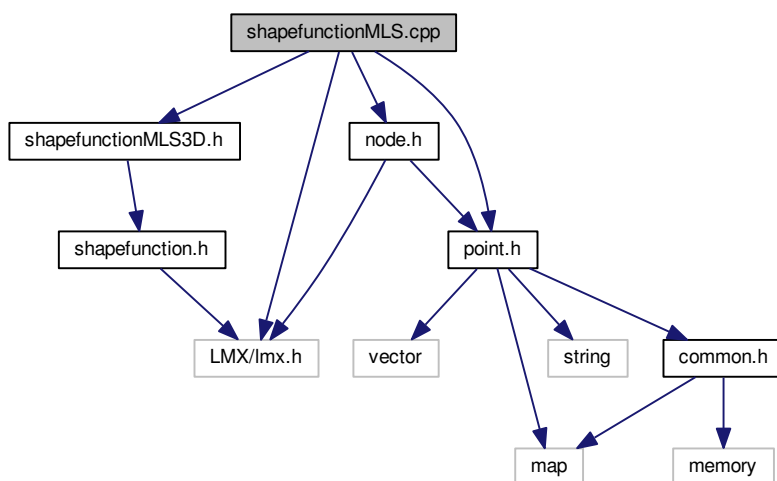
## Namespaces

- [mknix](#)

### 7.116 shapefunctionMLS.cpp File Reference

```
#include "shapefunctionMLS3D.h"  
#include "point.h"  
#include "node.h"  
#include "LMX/lmx.h"
```

Include dependency graph for shapefunctionMLS.cpp:



#### Namespaces

- [mknix](#)

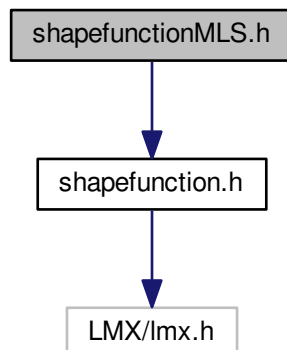
### 7.117 shapefunctionMLS.h File Reference

Function for aproximation of basic variables by Moving Least Squares fit.

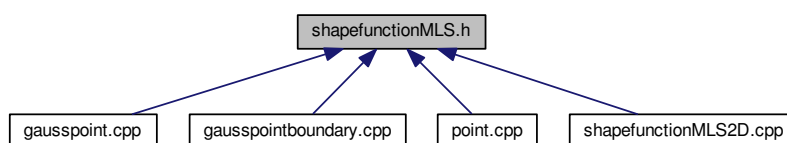


```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionMLS.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mnix::ShapeFunctionMLS](#)

## Namespaces

- [mnix](#)

### 7.117.1 Detailed Description

Function for aproximation of basic variables by Moving Least Squares fit.

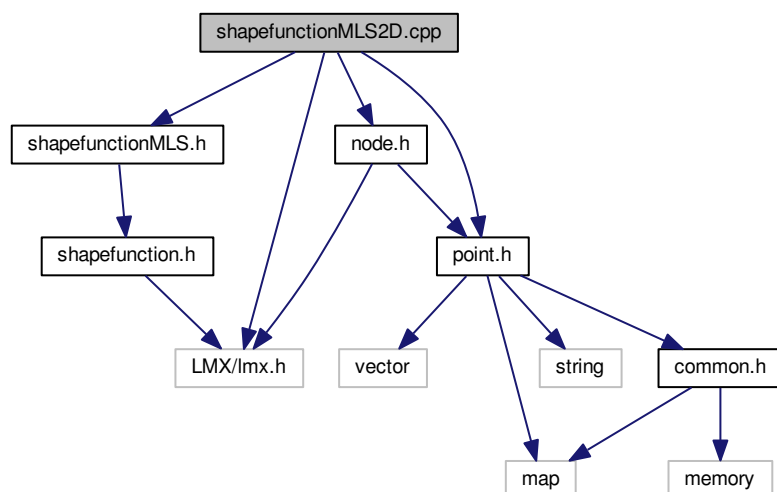
Author

Daniel Iglesias

### 7.118 shapefunctionMLS2D.cpp File Reference

```
#include "shapefunctionMLS.h"  
#include "point.h"  
#include "node.h"  
#include "LMX/lmx.h"
```

Include dependency graph for shapefunctionMLS2D.cpp:



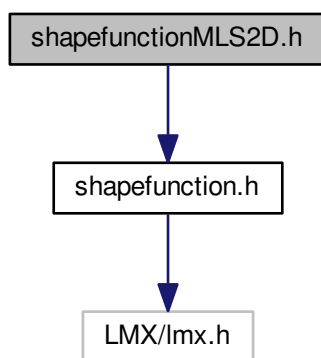
Namespaces

- [mknix](#)

## 7.119 shapefunctionMLS2D.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionMLS2D.h:



### Classes

- class [mknix::ShapeFunctionMLS](#)

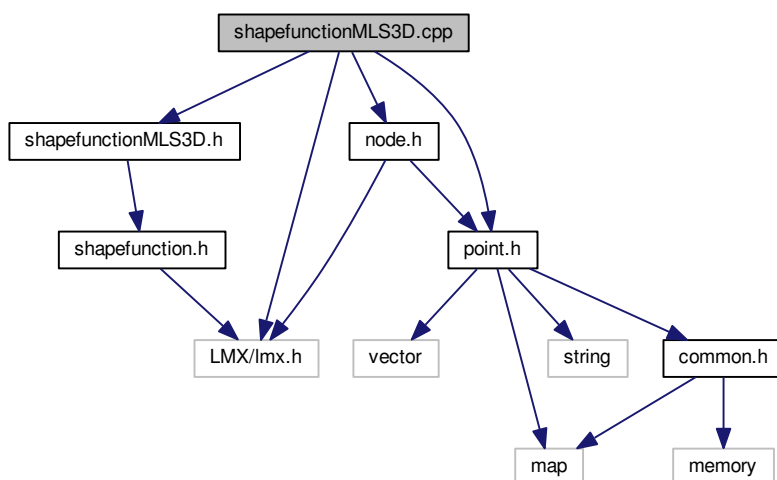
### Namespaces

- [mknix](#)

## 7.120 shapefunctionMLS3D.cpp File Reference

```
#include "shapefunctionMLS3D.h"  
#include "point.h"  
#include "node.h"  
#include "LMX/lmx.h"
```

Include dependency graph for shapefunctionMLS3D.cpp:



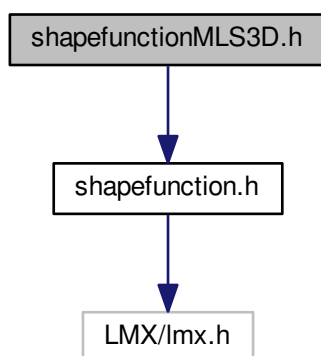
## Namespaces

- [mknix](#)

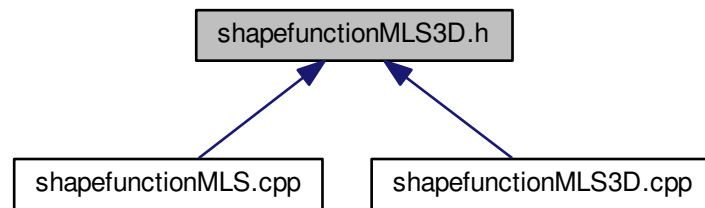
## 7.121 shapefunctionMLS3D.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionMLS3D.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::ShapeFunctionMLS`

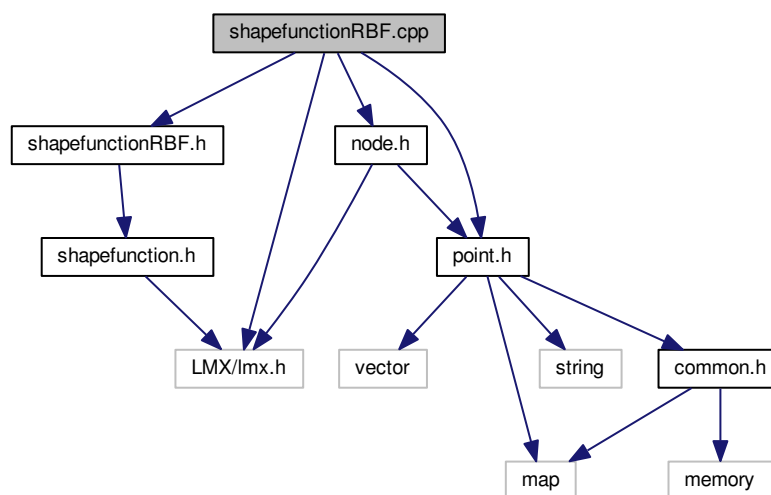
#### Namespaces

- `mnix`

## 7.122 shapefunctionRBF.cpp File Reference

```
#include "shapefunctionRBF.h"  
#include "point.h"  
#include "node.h"  
#include "LMX/lmx.h"
```

Include dependency graph for `shapefunctionRBF.cpp`:



## Namespaces

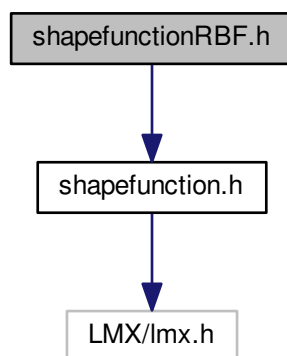
- [mknix](#)

### 7.123 shapefunctionRBF.h File Reference

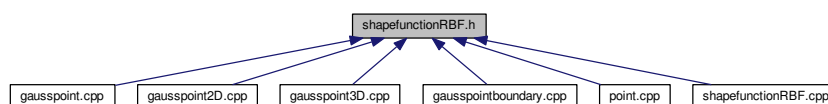
Function for interpolation of basic variables by means of Radial basis Functions.

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionRBF.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [mknix::ShapeFunctionRBF](#)

## Namespaces

- [mknix](#)

### 7.123.1 Detailed Description

Function for interpolation of basic variables by means of Radial basis Functions.

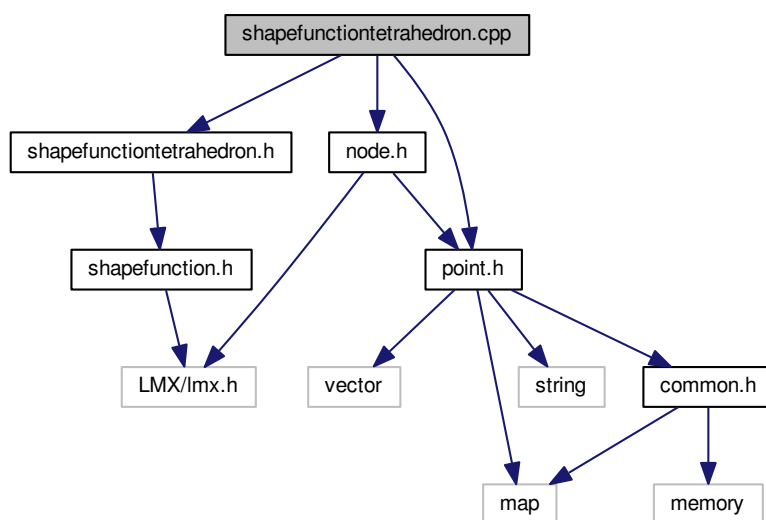
## Author

Daniel Iglesias

## 7.124 shapefunctiontetrahedron.cpp File Reference

```
#include "shapefunctiontetrahedron.h"  
#include "point.h"  
#include "node.h"
```

Include dependency graph for shapefunctiontetrahedron.cpp:



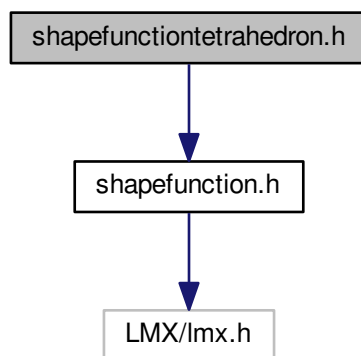
## Namespaces

- [mknix](#)

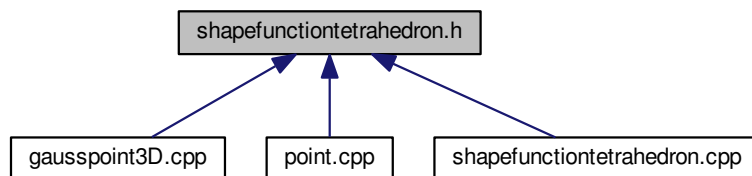
### 7.125 shapefunctiontetrahedron.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctiontetrahedron.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::ShapeFunctionTetrahedron](#)

#### Namespaces

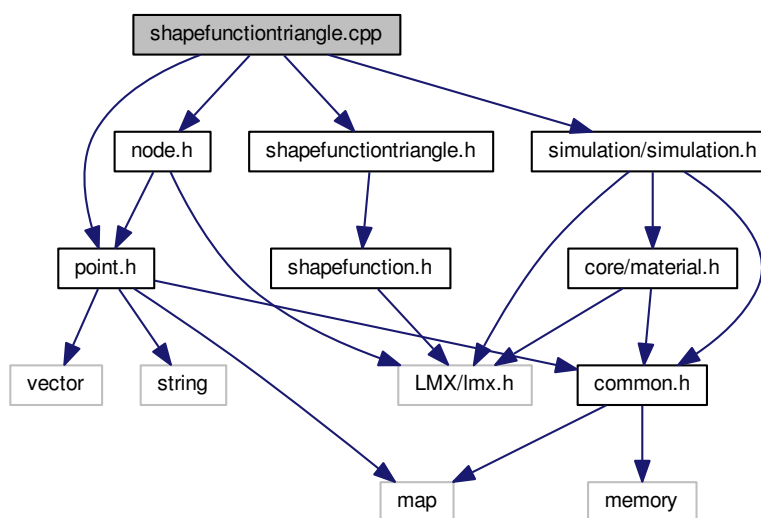
- [mknix](#)

### 7.126 shapefunctiontriangle.cpp File Reference

```
#include "shapefunctiontriangle.h"  
#include "point.h"  
#include "node.h"  
#include <simulation/simulation.h>
```



Include dependency graph for shapefunctiontriangle.cpp:



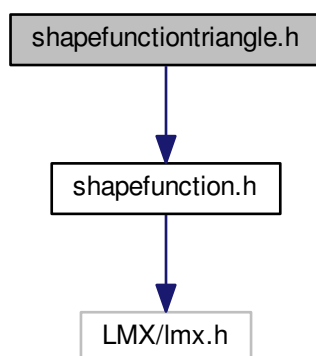
## Namespaces

- [mknix](#)

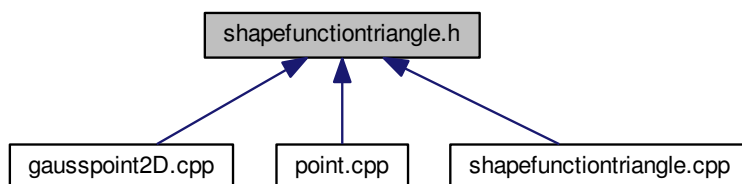
## 7.127 shapefunctiontriangle.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctiontriangle.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mknix::ShapeFunctionTriangle`

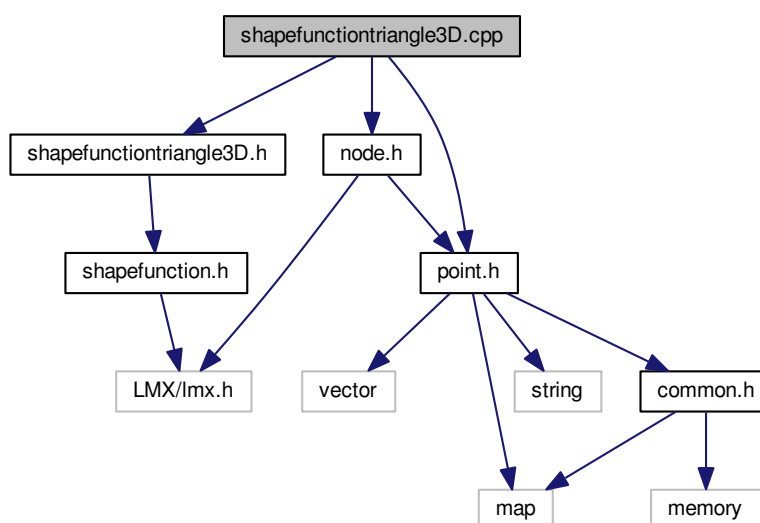
#### Namespaces

- `mknix`

### 7.128 shapefunctiontriangle3D.cpp File Reference

```
#include "shapefunctiontriangle3D.h"  
#include "point.h"  
#include "node.h"
```

Include dependency graph for `shapefunctiontriangle3D.cpp`:



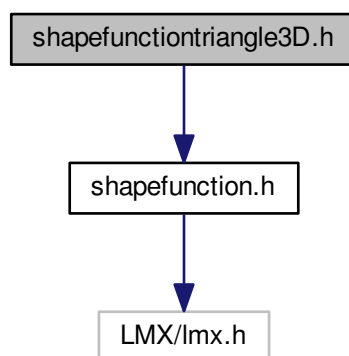
## Namespaces

- [mknix](#)

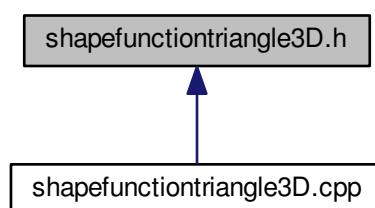
## 7.129 shapefunctiontriangle3D.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctiontriangle3D.h:



This graph shows which files directly or indirectly include this file:



## Classes

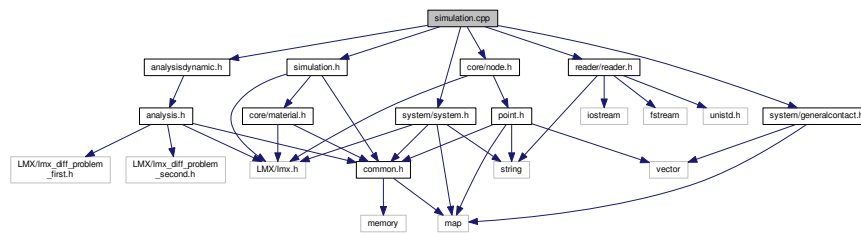
- class [mknix::ShapeFunctionTriangleSigned](#)

## Namespaces

- [mknix](#)

### 7.130 simulation.cpp File Reference

```
#include "simulation.h"
#include "analysisdynamic.h"
#include <system/system.h>
#include <core/node.h>
#include <reader/reader.h>
#include <system/generalcontact.h>
Include dependency graph for simulation.cpp:
```



### Namespaces

- [mknix](#)

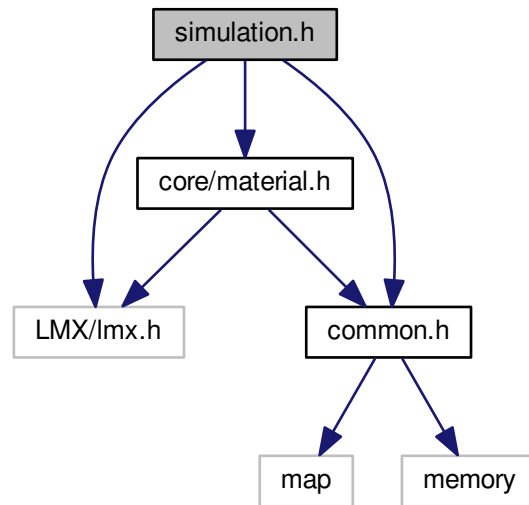
### Functions

- void [mknix::setSignal](#) (std::string node, std::vector< double >)
- std::vector< double > [mknix::getSignal](#) (std::string node)

### 7.131 simulation.h File Reference

```
#include "LMX/lmx.h"
#include "common.h"
#include <core/material.h>
```

Include dependency graph for simulation.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [mknix::Simulation](#)

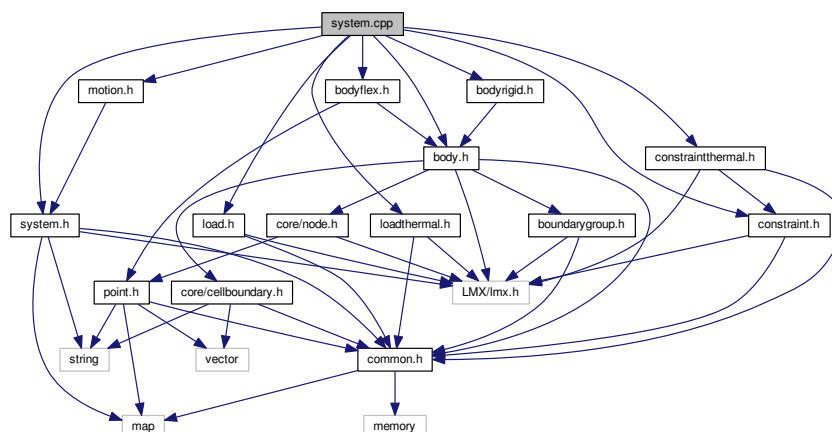
#### Namespaces

- [mknix](#)

## 7.132 system.cpp File Reference

```
#include "system.h"
#include "body.h"
#include "bodyflex.h"
#include "bodyrigid.h"
#include "constraint.h"
#include "constraintthermal.h"
#include "load.h"
#include "loadthermal.h"
#include "motion.h"
```

Include dependency graph for system.cpp:



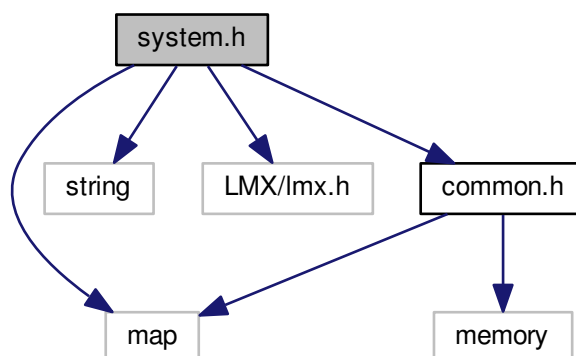
## Namespaces

- [mknix](#)

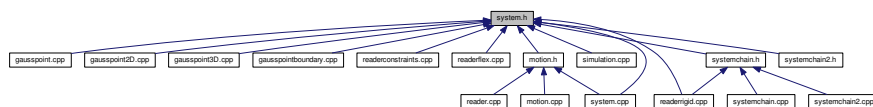
## 7.133 system.h File Reference

```
#include <map>
#include <string>
#include "LMX/lmx.h"
#include "common.h"
```

Include dependency graph for system.h:



This graph shows which files directly or indirectly include this file:



## Classes

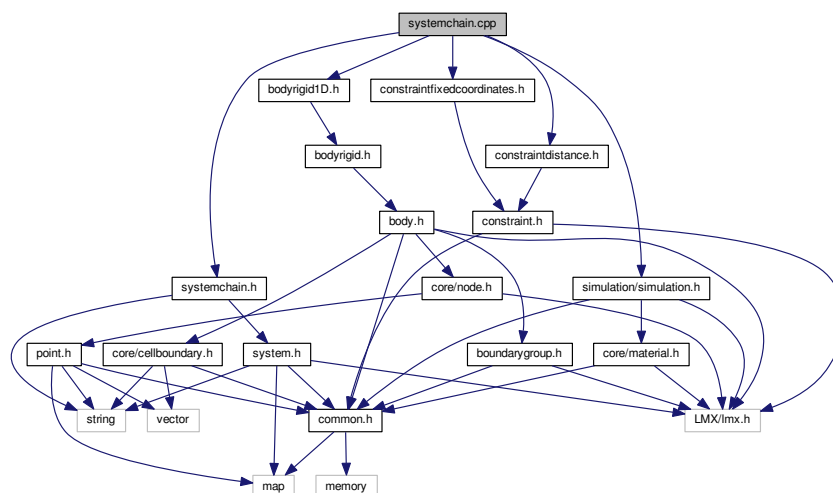
- class [mknix::System](#)

## Namespaces

- [mknix](#)

## 7.134 systemchain.cpp File Reference

```
#include "systemchain.h"
#include "bodyrigid1D.h"
#include "constraintdistance.h"
#include "constraintfixedcoordinates.h"
#include <simulation/simulation.h>
Include dependency graph for systemchain.cpp:
```



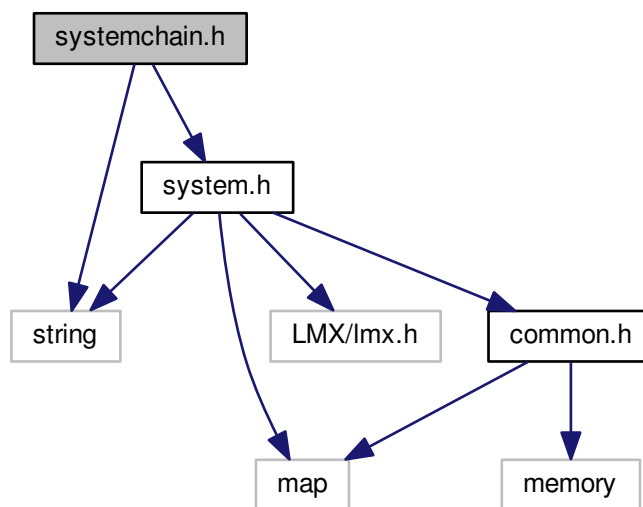
## Namespaces

- [mknix](#)

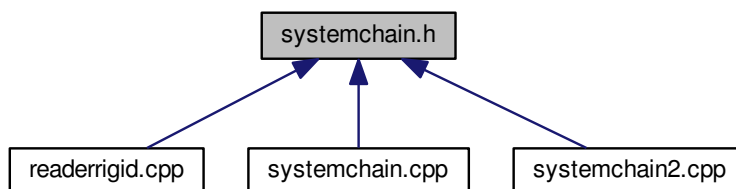
## 7.135 systemchain.h File Reference

```
#include "system.h"
#include <string>
```

Include dependency graph for systemchain.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `mnix::SystemChain`

#### Namespaces

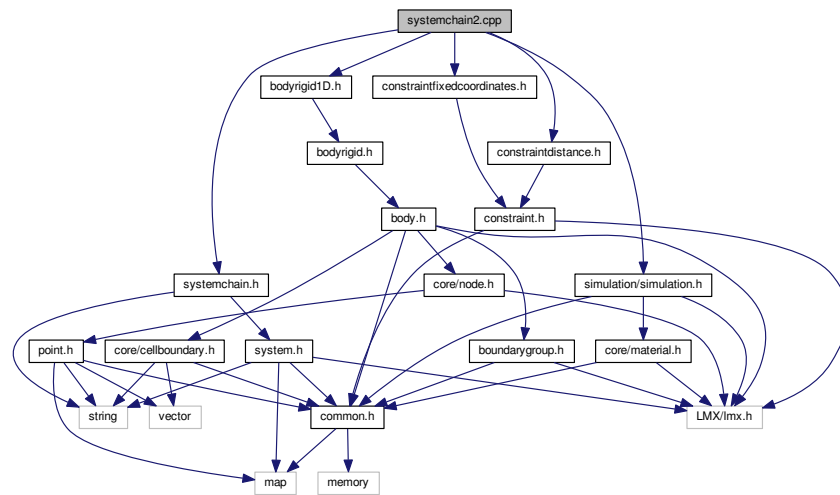
- `mnix`

### 7.136 systemchain2.cpp File Reference

```
#include "systemchain.h"
```



```
#include "bodyrigid1D.h"
#include "constraintdistance.h"
#include "constraintfixedcoordinates.h"
#include <simulation/simulation.h>
Include dependency graph for systemchain2.cpp:
```



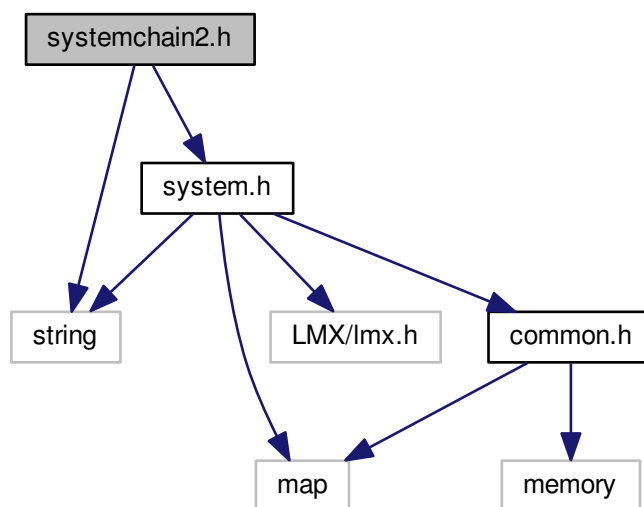
## Namespaces

- [mknix](#)

## 7.137 systemchain2.h File Reference

```
#include "system.h"
#include <string>
```

Include dependency graph for systemchain2.h:



#### Classes

- class [mknix::SystemChain](#)

#### Namespaces

- [mknix](#)

## Index

- ~Analysis
  - mknix::Analysis, [15](#)
- ~AnalysisDynamic
  - mknix::AnalysisDynamic, [18](#)
- ~AnalysisStatic
  - mknix::AnalysisStatic, [21](#)
- ~AnalysisThermalDynamic
  - mknix::AnalysisThermalDynamic, [23](#)
- ~AnalysisThermalStatic
  - mknix::AnalysisThermalStatic, [25](#)
- ~AnalysisThermoMechanicalDynamic
  - mknix::AnalysisThermoMechanicalDynamic, [27](#)
- ~Body
  - mknix::Body, [31](#)
- ~BoundaryGroup
  - mknix::BoundaryGroup, [39](#)
- ~Cell
  - mknix::Cell, [42](#)
- ~CellBoundary
  - mknix::CellBoundary, [46](#)
- ~CellBoundaryLinear
  - mknix::CellBoundaryLinear, [49](#)
- ~CellRect
  - mknix::CellRect, [52](#)
- ~CellTetrahedron
  - mknix::CellTetrahedron, [54](#)
- ~CellTriang
  - mknix::CellTriang, [58](#)
- ~CompBar
  - mknix::CompBar, [59](#)
- ~Constraint
  - mknix::Constraint, [61](#)
- ~ConstraintClearance
  - mknix::ConstraintClearance, [66](#)
- ~ConstraintContact
  - mknix::ConstraintContact, [69](#)
- ~ConstraintDistance
  - mknix::ConstraintDistance, [71](#)
- ~ConstraintFixedAxis
  - mknix::ConstraintFixedAxis, [75](#)
- ~ConstraintFixedCoordinates
  - mknix::ConstraintFixedCoordinates, [77](#)
- ~ConstraintThermal
  - mknix::ConstraintThermal, [80](#)
- ~ConstraintThermalFixed
  - mknix::ConstraintThermalFixed, [82](#)
- ~Contact
  - mknix::Contact, [84](#)
- ~ElemTetrahedron
  - mknix::ElemTetrahedron, [86](#)
- ~ElemTriangle
  - mknix::ElemTriangle, [89](#)
- ~FlexBody
  - mknix::FlexBody, [91](#)
- ~FlexFrameGalerkin
  - mknix::FlexFrameGalerkin, [98](#)
- ~FlexGlobalGalerkin
  - mknix::FlexGlobalGalerkin, [103](#)
- ~Force
  - mknix::Force, [107](#)
- ~GaussPoint
  - mknix::GaussPoint, [109](#)
- ~GaussPoint2D
  - mknix::GaussPoint2D, [116](#)
- ~GaussPoint3D
  - mknix::GaussPoint3D, [124](#)
- ~GaussPointBoundary
  - mknix::GaussPointBoundary, [132](#)
- ~Load
  - mknix::Load, [135](#)
- ~LoadThermal
  - mknix::LoadThermal, [136](#)
- ~LoadThermalBody
  - mknix::LoadThermalBody, [137](#)
- ~LoadThermalBoundary1D
  - mknix::LoadThermalBoundary1D, [139](#)
- ~Material
  - mknix::Material, [141](#)
- ~Motion
  - mknix::Motion, [148](#)
- ~Node
  - mknix::Node, [150](#)
- ~Point
  - mknix::Point, [159](#)
- ~Radiation
  - mknix::Radiation, [169](#)
- ~Reader
  - mknix::Reader, [170](#)
- ~ReaderConstraints
  - mknix::ReaderConstraints, [171](#)
- ~ReaderFlex
  - mknix::ReaderFlex, [172](#)
- ~ReaderRigid
  - mknix::ReaderRigid, [173](#)
- ~RigidBar
  - mknix::RigidBar, [175](#)
- ~RigidBody
  - mknix::RigidBody, [179](#)
- ~RigidBody2D
  - mknix::RigidBody2D, [184](#)
- ~RigidBody3D
  - mknix::RigidBody3D, [187](#)
- ~RigidBodyMassPoint
  - mknix::RigidBodyMassPoint, [190](#)
- ~ShapeFunction
  - mknix::ShapeFunction, [193](#)
- ~ShapeFunctionLinear
  - mknix::ShapeFunctionLinear, [197](#)
- ~ShapeFunctionLinearX
  - mknix::ShapeFunctionLinearX, [199](#)

- ~ShapeFunctionMLS
  - mknix::ShapeFunctionMLS, 201
- ~ShapeFunctionRBF
  - mknix::ShapeFunctionRBF, 203
- ~ShapeFunctionTetrahedron
  - mknix::ShapeFunctionTetrahedron, 205
- ~ShapeFunctionTriangle
  - mknix::ShapeFunctionTriangle, 207
- ~ShapeFunctionTriangleSigned
  - mknix::ShapeFunctionTriangleSigned, 208
- ~Simulation
  - mknix::Simulation, 211
- ~System
  - mknix::System, 230
- ~SystemChain
  - mknix::SystemChain, 236
- ~ThermalBody
  - mknix::ThermalBody, 239
- addBodyPoint
  - mknix::FlexBody, 91
- addBoundaryConnectivity
  - mknix::Body, 31
- addBoundaryGroup
  - mknix::Body, 31
- addBoundaryLine
  - mknix::Body, 31
- addCell
  - mknix::Body, 31
  - mknix::BoundaryGroup, 39
  - mknix::ThermalBody, 239
- addCellToBoundaryGroup
  - mknix::Body, 31
- addNode
  - mknix::Body, 31
  - mknix::BoundaryGroup, 39
  - mknix::RigidBar, 175
  - mknix::RigidBody2D, 184
  - mknix::RigidBody3D, 187
  - mknix::ThermalBody, 239
- addNodeToBoundaryGroup
  - mknix::Body, 32
- addNodes
  - mknix::Body, 31
- addPoint
  - mknix::FlexBody, 91, 92
- addSupportNode
  - mknix::Point, 159
- addThermalCapacity
  - mknix::Material, 141
- addThermalConductivity
  - mknix::Material, 141
- addTimeLenght
  - mknix::SystemChain, 236
- addToRender
  - mknix::CompBar, 59
- addVoxel
  - mknix::Radiation, 169
- addWeight
  - mknix::Node, 150
- alpha
  - mknix::Cell, 44
  - mknix::CellBoundary, 47
  - mknix::Constraint, 64
- alphaI
  - mknix::Point, 167
- Analysis
  - mknix::Analysis, 15
- analysis.cpp, 243
- analysis.h, 243
- AnalysisDynamic
  - mknix::AnalysisDynamic, 18
- AnalysisStatic
  - mknix::AnalysisStatic, 20
- AnalysisThermalDynamic
  - mknix::AnalysisThermalDynamic, 22
- AnalysisThermalStatic
  - mknix::AnalysisThermalStatic, 25
- AnalysisThermoMechanicalDynamic
  - mknix::AnalysisThermoMechanicalDynamic, 27
- analysisdynamic.cpp, 244
- analysisdynamic.h, 245
- analysisstatic.cpp, 246
- analysisstatic.h, 247
- analysisthermaldynamic.cpp, 247
- analysisthermaldynamic.h, 248
- analysisthermalstatic.cpp, 249
- analysisthermalstatic.h, 250
- analysisthermomechanicaldynamic.cpp, 250
- analysisthermomechanicaldynamic.h, 251
- assembleCapacityGaussPoints
  - mknix::Cell, 42
- assembleCapacityMatrix
  - mknix::Body, 32
  - mknix::System, 230
  - mknix::ThermalBody, 239
- assembleCij
  - mknix::GaussPoint, 110
- assembleConductivityGaussPoints
  - mknix::Cell, 42
- assembleConductivityMatrix
  - mknix::Body, 32
  - mknix::System, 230
  - mknix::ThermalBody, 239
- assembleConstraintForces
  - mknix::System, 230
- assembleExternalForces
  - mknix::Body, 32
  - mknix::FlexFrameGalerkin, 98
  - mknix::FlexGlobalGalerkin, 103
  - mknix::Load, 135
  - mknix::RigidBody, 179
  - mknix::System, 230
- assembleExternalHeat
  - mknix::Body, 32
  - mknix::BoundaryGroup, 39
  - mknix::LoadThermal, 136

- mknix::System, [230](#)
  - mknix::ThermalBody, [240](#)
- assembleFext
  - mknix::GaussPoint, [110](#)
  - mknix::GaussPoint2D, [116](#)
  - mknix::GaussPoint3D, [124](#)
- assembleFextGaussPoints
  - mknix::Cell, [42](#)
- assembleFint
  - mknix::GaussPoint, [110](#)
  - mknix::GaussPoint2D, [116](#)
  - mknix::GaussPoint3D, [124](#)
- assembleFintGaussPoints
  - mknix::Cell, [42](#)
- assembleHij
  - mknix::GaussPoint, [110](#)
- assembleInternalForces
  - mknix::Constraint, [61](#)
  - mknix::ConstraintThermal, [80](#)
  - mknix::FlexBody, [92](#)
  - mknix::FlexFrameGalerkin, [98](#)
  - mknix::FlexGlobalGalerkin, [103](#)
  - mknix::System, [230](#)
- assembleInternalHeat
  - mknix::System, [230](#)
- assembleKGaussPoints
  - mknix::Cell, [42](#)
- assembleKij
  - mknix::GaussPoint, [110](#)
  - mknix::GaussPoint2D, [116](#)
  - mknix::GaussPoint3D, [124](#)
- assembleMGaussPoints
  - mknix::Cell, [42](#)
- assembleMassMatrix
  - mknix::Body, [32](#)
  - mknix::FlexFrameGalerkin, [98](#)
  - mknix::FlexGlobalGalerkin, [103](#)
  - mknix::RigidBody, [179](#)
  - mknix::System, [230](#)
- assembleMij
  - mknix::GaussPoint, [110](#)
  - mknix::GaussPoint2D, [116](#)
  - mknix::GaussPoint3D, [124](#)
- assembleNLRGaussPoints
  - mknix::Cell, [42](#)
- assembleQext
  - mknix::GaussPoint, [110](#)
  - mknix::GaussPointBoundary, [132](#)
- assembleQextGaussPoints
  - mknix::Cell, [43](#)
  - mknix::CellBoundary, [46](#)
- assembleRGaussPoints
  - mknix::Cell, [43](#)
- assembleRi
  - mknix::GaussPoint, [110](#)
  - mknix::GaussPoint2D, [116](#)
  - mknix::GaussPoint3D, [124](#)
- assembleTangentMatrix
  - mknix::Constraint, [62](#)
  - mknix::ConstraintThermal, [80](#)
  - mknix::FlexBody, [92](#)
  - mknix::FlexFrameGalerkin, [98](#)
  - mknix::FlexGlobalGalerkin, [104](#)
  - mknix::System, [230](#)
- assembleThermalTangentMatrix
  - mknix::System, [230](#)
- axisName
  - mknix::ConstraintFixedAxis, [75](#)
- B
  - mknix::GaussPoint, [113](#)
- Body
  - mknix::Body, [30](#)
- body.cpp, [252](#)
- body.h, [253](#)
- bodyPoints
  - mknix::Cell, [44](#)
  - mknix::CellBoundary, [47](#)
  - mknix::FlexBody, [95](#)
- bodyflex.cpp, [254](#)
- bodyflex.h, [254](#)
- bodyflexframegalerkin.cpp, [255](#)
- bodyflexframegalerkin.h, [256](#)
- bodyflexglobalgalerkin.cpp, [258](#)
- bodyflexglobalgalerkin.h, [258](#)
- bodyrigid.cpp, [260](#)
- bodyrigid.h, [260](#)
- bodyrigid0D.cpp, [261](#)
- bodyrigid0D.h, [262](#)
- bodyrigid1D.cpp, [264](#)
- bodyrigid1D.h, [264](#)
- bodyrigid2D.cpp, [266](#)
- bodyrigid2D.h, [266](#)
- bodyrigid3D.cpp, [268](#)
- bodyrigid3D.h, [268](#)
- bodythermal.cpp, [270](#)
- bodythermal.h, [270](#)
- bondedBodyNodes
  - mknix::Body, [37](#)
- boundaryConnectivity
  - mknix::Body, [37](#)
- BoundaryGroup
  - mknix::BoundaryGroup, [39](#)
- boundaryGroups
  - mknix::Body, [37](#)
- boundarygroup.cpp, [271](#)
- boundarygroup.h, [272](#)
- C
  - mknix::GaussPoint, [113](#)
- calc
  - mknix::ShapeFunction, [193](#)
  - mknix::ShapeFunctionLinear, [197](#)
  - mknix::ShapeFunctionLinearX, [199](#)
  - mknix::ShapeFunctionMLS, [201](#)
  - mknix::ShapeFunctionRBF, [203](#)
  - mknix::ShapeFunctionTetrahedron, [205](#)

- mknix::ShapeFunctionTriangle, 207
  - mknix::ShapeFunctionTriangleSigned, 209
- calcCapacityMatrix
  - mknix::Body, 32
  - mknix::System, 230
  - mknix::ThermalBody, 240
- calcConductivityMatrix
  - mknix::Body, 32
  - mknix::System, 230
  - mknix::ThermalBody, 240
- calcElasticE
  - mknix::GaussPoint, 110
  - mknix::GaussPoint2D, 117
  - mknix::GaussPoint3D, 125
- calcElasticEGaussPoints
  - mknix::Cell, 43
- calcExternalForces
  - mknix::Body, 33
  - mknix::FlexFrameGalerkin, 98
  - mknix::FlexGlobalGalerkin, 104
  - mknix::RigidBar, 176
  - mknix::RigidBody2D, 184
  - mknix::RigidBody3D, 187
  - mknix::RigidBodyMassPoint, 190
  - mknix::System, 231
- calcExternalHeat
  - mknix::Body, 33
  - mknix::BoundaryGroup, 39
  - mknix::System, 231
  - mknix::ThermalBody, 240
- calcInternalForces
  - mknix::Constraint, 62
  - mknix::FlexBody, 92
  - mknix::FlexFrameGalerkin, 99
  - mknix::FlexGlobalGalerkin, 104
  - mknix::System, 231
- calcInternalHeat
  - mknix::System, 231
- calcKineticE
  - mknix::GaussPoint, 110
  - mknix::GaussPoint2D, 117
  - mknix::GaussPoint3D, 125
- calcKineticEGaussPoints
  - mknix::Cell, 43
- calcMassMatrix
  - mknix::Body, 33
  - mknix::FlexFrameGalerkin, 99
  - mknix::FlexGlobalGalerkin, 104
  - mknix::RigidBar, 176
  - mknix::RigidBody2D, 185
  - mknix::RigidBody3D, 188
  - mknix::RigidBodyMassPoint, 190
  - mknix::System, 231
- calcPhi
  - mknix::Constraint, 62
  - mknix::ConstraintClearance, 67
  - mknix::ConstraintContact, 69
  - mknix::ConstraintDistance, 72
  - mknix::ConstraintFixedAxis, 75
  - mknix::ConstraintFixedCoordinates, 77
  - mknix::ConstraintThermalFixed, 82
- calcPhiq
  - mknix::Constraint, 63
  - mknix::ConstraintClearance, 67
  - mknix::ConstraintContact, 69
  - mknix::ConstraintDistance, 72
  - mknix::ConstraintFixedAxis, 75
  - mknix::ConstraintFixedCoordinates, 77
  - mknix::ConstraintThermalFixed, 82
- calcPhiqq
  - mknix::Constraint, 63
  - mknix::ConstraintClearance, 67
  - mknix::ConstraintContact, 69
  - mknix::ConstraintDistance, 72
  - mknix::ConstraintFixedAxis, 75
  - mknix::ConstraintFixedCoordinates, 78
  - mknix::ConstraintThermalFixed, 83
- calcPotentialE
  - mknix::GaussPoint, 110
  - mknix::GaussPoint2D, 117
  - mknix::GaussPoint3D, 125
- calcPotentialEGaussPoints
  - mknix::Cell, 43
- calcRo
  - mknix::ConstraintDistance, 72
- calcTangentMatrix
  - mknix::Constraint, 63
  - mknix::FlexBody, 92
  - mknix::FlexFrameGalerkin, 99
  - mknix::FlexGlobalGalerkin, 104
  - mknix::System, 231
- calcThermalTangentMatrix
  - mknix::System, 231
- Cell
  - mknix::Cell, 42
- cell.cpp, 273
- cell.h, 274
- CellBoundary
  - mknix::CellBoundary, 46
- CellBoundaryLinear
  - mknix::CellBoundaryLinear, 49
- CellRect
  - mknix::CellRect, 52
- CellTetrahedron
  - mknix::CellTetrahedron, 54
- CellTriang
  - mknix::CellTriang, 57
- cellboundary.cpp, 275
- cellboundary.h, 275
- cellboundarylinear.cpp, 276
- cellboundarylinear.h, 277
- cellrect.cpp, 278
- cellrect.h, 279
- cells
  - mknix::Body, 37
  - mknix::BoundaryGroup, 40

- mknix::ThermalBody, [242](#)
- celltetrahedron.cpp, [281](#)
- celltetrahedron.h, [281](#)
- celltriang.cpp, [283](#)
- celltriang.h, [283](#)
- checkAugmented
  - mknix::Constraint, [64](#)
  - mknix::System, [231](#)
- clearAugmented
  - mknix::Constraint, [64](#)
  - mknix::System, [231](#)
- common.cpp, [285](#)
- common.h, [285](#)
- CompBar
  - mknix::CompBar, [59](#)
- compbar.cpp, [286](#)
- compbar.h, [286](#)
- compute\_abcd
  - mknix::ShapeFunctionTetrahedron, [205](#)
- computeC
  - mknix::Material, [141](#)
- computeCapacityGaussPoints
  - mknix::Cell, [43](#)
- computeCij
  - mknix::GaussPoint, [110](#)
- computeConductivityGaussPoints
  - mknix::Cell, [43](#)
- computeD
  - mknix::Material, [141](#)
- computeEnergy
  - mknix::Body, [37](#)
  - mknix::FlexBody, [95](#)
  - mknix::Material, [142](#)
  - mknix::RigidBody, [182](#)
  - mknix::ThermalBody, [242](#)
- computeFext
  - mknix::GaussPoint, [111](#)
  - mknix::GaussPoint2D, [118](#)
  - mknix::GaussPoint3D, [126](#)
- computeFextGaussPoints
  - mknix::Cell, [43](#)
- computeFint
  - mknix::GaussPoint, [111](#)
  - mknix::GaussPoint2D, [118](#)
  - mknix::GaussPoint3D, [126](#)
- computeFintGaussPoints
  - mknix::Cell, [43](#)
- computeHij
  - mknix::GaussPoint, [111](#)
- computeKGaussPoints
  - mknix::Cell, [43](#)
- computeKij
  - mknix::GaussPoint, [111](#)
  - mknix::GaussPoint2D, [118](#)
  - mknix::GaussPoint3D, [126](#)
- computeMGaussPoints
  - mknix::Cell, [43](#)
- computeMij
  - mknix::GaussPoint, [111](#)
  - mknix::GaussPoint2D, [119](#)
  - mknix::GaussPoint3D, [127](#)
- computeNLFint
  - mknix::GaussPoint, [111](#)
  - mknix::GaussPoint2D, [119](#)
  - mknix::GaussPoint3D, [127](#)
- computeNLFintGaussPoints
  - mknix::Cell, [43](#)
- computeNLKGaussPoints
  - mknix::Cell, [43](#)
- computeNLKij
  - mknix::GaussPoint, [111](#)
  - mknix::GaussPoint2D, [120](#)
  - mknix::GaussPoint3D, [128](#)
- computeNLStress
  - mknix::GaussPoint, [111](#)
  - mknix::GaussPoint2D, [120](#)
  - mknix::GaussPoint3D, [128](#)
- computeQext
  - mknix::GaussPoint, [111](#)
  - mknix::GaussPointBoundary, [132](#)
- computeQextGaussPoints
  - mknix::Cell, [43](#)
  - mknix::CellBoundary, [46](#)
- computeS
  - mknix::Material, [142](#), [143](#)
- computeShapeFunctions
  - mknix::Cell, [43](#)
  - mknix::CellBoundary, [46](#)
  - mknix::CellBoundaryLinear, [50](#)
  - mknix::ElemTetrahedron, [86](#)
  - mknix::ElemTriangle, [89](#)
- computeStress
  - mknix::FlexBody, [95](#)
  - mknix::GaussPoint, [112](#)
  - mknix::GaussPoint2D, [121](#)
  - mknix::GaussPoint3D, [129](#)
- Constraint
  - mknix::Constraint, [61](#)
- constraint.cpp, [287](#)
- constraint.h, [287](#)
- ConstraintClearance
  - mknix::ConstraintClearance, [66](#)
- ConstraintContact
  - mknix::ConstraintContact, [69](#)
- ConstraintDistance
  - mknix::ConstraintDistance, [71](#), [72](#)
- ConstraintFixedAxis
  - mknix::ConstraintFixedAxis, [74](#)
- ConstraintFixedCoordinates
  - mknix::ConstraintFixedCoordinates, [77](#)
- ConstraintThermal
  - mknix::ConstraintThermal, [80](#)
- ConstraintThermalFixed
  - mknix::ConstraintThermalFixed, [82](#)
- constraintclearance.cpp, [288](#)
- constraintclearance.h, [289](#)

- constraintcontact.cpp, 290
- constraintcontact.h, 291
- constraintdistance.cpp, 292
- constraintdistance.h, 292
- constraintfixedaxis.cpp, 293
- constraintfixedaxis.h, 294
- constraintfixedcoordinates.cpp, 295
- constraintfixedcoordinates.h, 296
- constraints
  - mknix::System, 233
- constraintsThermal
  - mknix::System, 233
- constraintthermal.cpp, 297
- constraintthermal.h, 297
- constraintthermalfixed.cpp, 298
- constraintthermalfixed.h, 299
- Contact
  - mknix::Contact, 84
  - mknix::Simulation, 227
  - mknix::System, 233
- createDelaunay
  - mknix::Contact, 84
- createDrawingContactObjects
  - mknix::Contact, 84
- createDrawingObjects
  - mknix::Contact, 84
- createGaussPoints
  - mknix::CellBoundaryLinear, 50
  - mknix::CellTetrahedron, 55
  - mknix::CellTriang, 58
- createGaussPoints\_MC
  - mknix::ElemTetrahedron, 86
  - mknix::ElemTriangle, 89
- createPoints
  - mknix::Contact, 84
- createPolys
  - mknix::Contact, 84
- data\_type
  - mknix, 12
- dc
  - mknix::Cell, 44
  - mknix::CellBoundary, 47
  - mknix::Point, 167
- densityFactor
  - mknix::RigidBody, 182
- dictionary.h, 301
- dim
  - mknix::Constraint, 64
  - mknix::Point, 167
  - mknix::RigidBody, 182
  - mknix::ShapeFunction, 195
- distance
  - mknix::Point, 159
- domainConf
  - mknix::RigidBody, 182
- drawObjects
  - mknix::Contact, 84
- dynamicAcceleration
  - mknix::Simulation, 211
- dynamicConvergence
  - mknix::Simulation, 211
- dynamicResidue
  - mknix::Simulation, 212
- dynamicTangent
  - mknix::Simulation, 212
- dynamicThermalConvergence
  - mknix::Simulation, 213
- dynamicThermalConvergenceInThermomechanical
  - mknix::Simulation, 213
- dynamicThermalEvaluation
  - mknix::Simulation, 213
- dynamicThermalResidue
  - mknix::Simulation, 214
- dynamicThermalTangent
  - mknix::Simulation, 214
- ElemTetrahedron
  - mknix::ElemTetrahedron, 86
- ElemTriangle
  - mknix::ElemTriangle, 88
- elemtetrahedron.cpp, 301
- elemtetrahedron.h, 301
- elemtriangle.cpp, 303
- elemtriangle.h, 303
- endSimulation
  - mknix::Simulation, 214
- energies
  - mknix::FlexBody, 96
- energy
  - mknix::RigidBody, 182
- epsilon
  - mknix::Analysis, 16
- explicitAcceleration
  - mknix::Simulation, 214
- explicitThermalEvaluation
  - mknix::Simulation, 215
- externalForces
  - mknix::Load, 135
  - mknix::RigidBody, 182
- externalHeat
  - mknix::Load, 135
  - mknix::LoadThermal, 137
- fext
  - mknix::GaussPoint, 113
- fillFEmatrices
  - mknix::GaussPoint, 112
  - mknix::GaussPoint2D, 121
  - mknix::GaussPoint3D, 129
- findSupportNodes
  - mknix::Point, 160
- fint
  - mknix::GaussPoint, 113
- flexBodies
  - mknix::System, 233
- FlexBody
  - mknix::FlexBody, 91



- FlexFrameGalerkin
  - mknix::FlexFrameGalerkin, [98](#)
- FlexGlobalGalerkin
  - mknix::FlexGlobalGalerkin, [103](#)
- Force
  - mknix::Force, [107](#)
- force.cpp, [305](#)
- force.h, [305](#)
- formulation
  - mknix::Cell, [44](#)
  - mknix::CellBoundary, [47](#)
  - mknix::FlexBody, [96](#)
- frameNodes
  - mknix::RigidBody, [182](#)
- gPoints
  - mknix::Cell, [44](#)
  - mknix::CellBoundary, [47](#)
- gPoints\_MC
  - mknix::Cell, [44](#)
- GaussPoint
  - mknix::GaussPoint, [109](#)
  - mknix::Point, [166](#)
- GaussPoint2D
  - mknix::GaussPoint2D, [116](#)
- GaussPoint3D
  - mknix::GaussPoint3D, [124](#)
- GaussPointBoundary
  - mknix::GaussPointBoundary, [132](#)
- gausspoint.cpp, [306](#)
- gausspoint.h, [307](#)
- gausspoint2D.cpp, [308](#)
- gausspoint2D.h, [309](#)
- gausspoint3D.cpp, [310](#)
- gausspoint3D.h, [310](#)
- gausspointboundary.cpp, [312](#)
- gausspointboundary.h, [312](#)
- generalcontact.cpp, [313](#)
- generalcontact.h, [313](#)
- getAlpha
  - mknix::Simulation, [215](#)
- getBodyPoint
  - mknix::FlexBody, [92](#)
- getBoundaryFirstNode
  - mknix::Body, [33](#)
- getBoundaryNextNode
  - mknix::Body, [33](#)
- getBoundarySize
  - mknix::Body, [33](#)
- getC
  - mknix::Material, [143](#)
- getCapacity
  - mknix::Material, [143](#)
- getCellLastNumber
  - mknix::Body, [34](#)
- getConf
  - mknix::Node, [150](#)
  - mknix::Point, [160](#)
- getConstraintMethod
  - mknix::Simulation, [216](#)
- getCsym
  - mknix::Material, [144](#)
- getD
  - mknix::Material, [144](#)
- getDensity
  - mknix::Material, [144](#)
- getDim
  - mknix::Point, [161](#)
  - mknix::Simulation, [216](#)
- getE
  - mknix::Material, [145](#)
- getGap
  - mknix::ConstraintContact, [69](#)
- getGravity
  - mknix::Simulation, [217](#)
- getInterfaceNodesCoords
  - mknix::Simulation, [218](#)
- getInternalForces
  - mknix::ConstraintClearance, [67](#)
  - mknix::ConstraintContact, [69](#)
  - mknix::ConstraintDistance, [72](#)
  - mknix::ConstraintFixedAxis, [75](#)
  - mknix::ConstraintFixedCoordinates, [78](#)
  - mknix::ConstraintThermalFixed, [83](#)
- getKappa
  - mknix::Material, [145](#)
- getLastBodyPoint
  - mknix::FlexBody, [92](#)
- getLastNode
  - mknix::Body, [34](#)
  - mknix::ThermalBody, [240](#)
- getLoadThermalBody
  - mknix::LoadThermalBody, [138](#)
- getLoadThermalBoundary1D
  - mknix::LoadThermalBoundary1D, [139](#)
- getMaxTemp
  - mknix::LoadThermal, [136](#)
- getMu
  - mknix::Material, [146](#)
- getNode
  - mknix::Body, [34](#)
  - mknix::Constraint, [64](#)
  - mknix::FlexBody, [92](#)
  - mknix::RigidBody, [180](#)
  - mknix::System, [231](#)
  - mknix::SystemChain, [236](#)
  - mknix::ThermalBody, [240](#)
- getNodes
  - mknix::Body, [34](#)
- getNodesSize
  - mknix::Body, [34](#)
- getNumber
  - mknix::Point, [161](#)
- getNumberOfPoints
  - mknix::FlexBody, [93](#)
- getOutputSignalThermal
  - mknix::System, [231](#)

- getPhi
  - mknix::ShapeFunction, 193
- getShapeFunType
  - mknix::Point, 161
- getShapeFunValue
  - mknix::Node, 152
- getSignal
  - mknix, 13
- getSmoothingType
  - mknix::Simulation, 218
- getSparsePattern
  - mknix::Simulation, 219
- getStiffnessMatrix
  - mknix::ConstraintClearance, 67
  - mknix::ConstraintContact, 69
  - mknix::ConstraintDistance, 73
  - mknix::ConstraintFixedAxis, 75
  - mknix::ConstraintFixedCoordinates, 78
  - mknix::ConstraintThermalFixed, 83
- getSupportNodeNumber
  - mknix::Node, 152
  - mknix::Point, 161
- getSupportNodes
  - mknix::Point, 162
- getSupportSize
  - mknix::Node, 152
  - mknix::Point, 162
- getTemp
  - mknix::Node, 153
  - mknix::Point, 162
- getThermalNodes
  - mknix::System, 231
- getThermalNumber
  - mknix::Node, 153
- getTime
  - mknix::Simulation, 220
- getTitle
  - mknix::System, 231
- getType
  - mknix::Body, 34
  - mknix::FlexFrameGalerkin, 99
  - mknix::FlexGlobalGalerkin, 104
  - mknix::RigidBar, 176
  - mknix::RigidBody2D, 185
  - mknix::RigidBody3D, 188
  - mknix::RigidBodyMassPoint, 190
- getU
  - mknix::Node, 153
- getUx
  - mknix::Node, 154
- getUy
  - mknix::Node, 154
- getUz
  - mknix::Node, 154
- getWeight
  - mknix::Node, 155
- getX
  - mknix::Point, 162
- getY
  - mknix::Point, 163
- getZ
  - mknix::Point, 164
- getqt
  - mknix::Node, 151
- getqx
  - mknix::Node, 152
- gnuplotOut
  - mknix::Cell, 44
  - mknix::CellRect, 52
  - mknix::CellTetrahedron, 55
  - mknix::CellTriang, 58
  - mknix::Point, 164
  - mknix::ShapeFunction, 194
- gnuplotOutStress
  - mknix::Cell, 44
  - mknix::GaussPoint, 112
- gp
  - mknix::ShapeFunction, 195
- groundNodes
  - mknix::System, 233
- H
  - mknix::GaussPoint, 113
- init
  - mknix::Analysis, 15
  - mknix::AnalysisThermalDynamic, 23
  - mknix::Simulation, 220
- initFlexBodies
  - mknix::System, 232
- initialize
  - mknix::Body, 34
  - mknix::BoundaryGroup, 39
  - mknix::Cell, 44
  - mknix::CellBoundary, 46
  - mknix::CellBoundaryLinear, 50
  - mknix::CellRect, 52
  - mknix::ElemTetrahedron, 86
  - mknix::ElemTriangle, 89
  - mknix::FlexBody, 93
  - mknix::ThermalBody, 241
- initializeMatVecs
  - mknix::GaussPointBoundary, 132
- inputFromFile
  - mknix::Reader, 170
  - mknix::Simulation, 220
- insertNodesXCoordinates
  - mknix::LoadThermal, 136
- internalForces
  - mknix::Constraint, 64
- internalForcesOutput
  - mknix::Constraint, 64
- interpolate1D
  - mknix, 13
- isThermal
  - mknix::Body, 37

- jacobian
  - mknix::Cell, [44](#)
  - mknix::CellBoundary, [47](#)
  - mknix::Point, [167](#)
- K
  - mknix::GaussPoint, [113](#)
- lambda
  - mknix::Constraint, [65](#)
- lastNode
  - mknix::Body, [37](#)
- linearBoundary
  - mknix::Body, [37](#)
- Imx, [11](#)
- Imx::DenseMatrix< T >, [84](#)
- Imx::Matrix< T >, [146](#)
- Imx::Vector< T >, [242](#)
- Load
  - mknix::Load, [135](#)
- load.cpp, [314](#)
- load.h, [315](#)
- loadFile
  - mknix::LoadThermalBoundary1D, [140](#)
- LoadThermal
  - mknix::LoadThermal, [136](#)
- LoadThermalBody
  - mknix::LoadThermalBody, [137](#)
- loadThermalBody
  - mknix::Body, [37](#)
  - mknix::ThermalBody, [242](#)
- LoadThermalBoundary1D
  - mknix::LoadThermalBoundary1D, [139](#)
- loadThermalBoundaryGroup
  - mknix::BoundaryGroup, [40](#)
- loadTimeFile
  - mknix::LoadThermalBoundary1D, [140](#)
- loadmap
  - mknix::LoadThermalBoundary1D, [140](#)
- loadradiation.cpp, [316](#)
- loadradiation.h, [316](#)
- loads
  - mknix::System, [233](#)
- loadsThermal
  - mknix::System, [234](#)
- loadthermal.cpp, [317](#)
- loadthermal.h, [318](#)
- loadthermalbody.cpp, [319](#)
- loadthermalbody.h, [320](#)
- loadthermalboundary1D.cpp, [320](#)
- loadthermalboundary1D.h, [321](#)
- localMassMatrix
  - mknix::RigidBody, [182](#)
- M
  - mknix::GaussPoint, [113](#)
- make\_unique
  - mknix, [13](#)
- mass
  - mknix::RigidBody, [182](#)
- mat
  - mknix::Cell, [44](#)
  - mknix::GaussPoint, [113](#)
- Material
  - mknix::Material, [141](#)
- material.cpp, [322](#)
- material.h, [323](#)
- method
  - mknix::Constraint, [65](#)
- mknix, [11](#)
  - data\_type, [12](#)
  - getSignal, [13](#)
  - interpolate1D, [13](#)
  - make\_unique, [13](#)
  - setSignal, [13](#)
- mknix::Analysis, [13](#)
  - ~Analysis, [15](#)
  - Analysis, [15](#)
  - epsilon, [16](#)
  - init, [15](#)
  - nextStep, [15](#)
  - setEpsilon, [15](#)
  - solve, [16](#)
  - theSimulation, [16](#)
  - type, [16](#)
- mknix::AnalysisDynamic, [17](#)
  - ~AnalysisDynamic, [18](#)
  - AnalysisDynamic, [18](#)
  - solve, [18](#)
  - type, [19](#)
- mknix::AnalysisStatic, [19](#)
  - ~AnalysisStatic, [21](#)
  - AnalysisStatic, [20](#)
  - solve, [21](#)
  - type, [21](#)
- mknix::AnalysisThermalDynamic, [21](#)
  - ~AnalysisThermalDynamic, [23](#)
  - AnalysisThermalDynamic, [22](#)
  - init, [23](#)
  - nextStep, [23](#)
  - solve, [23](#)
  - type, [23](#)
- mknix::AnalysisThermalStatic, [24](#)
  - ~AnalysisThermalStatic, [25](#)
  - AnalysisThermalStatic, [25](#)
  - solve, [25](#)
  - type, [25](#)
- mknix::AnalysisThermoMechanicalDynamic, [25](#)
  - ~AnalysisThermoMechanicalDynamic, [27](#)
  - AnalysisThermoMechanicalDynamic, [27](#)
  - solve, [27](#)
  - type, [28](#)
- mknix::Body, [28](#)
  - ~Body, [31](#)
  - addBoundaryConnectivity, [31](#)
  - addBoundaryGroup, [31](#)
  - addBoundaryLine, [31](#)

- addCell, 31
- addCellToBoundaryGroup, 31
- addNode, 31
- addNodeToBoundaryGroup, 32
- addNodes, 31
- assembleCapacityMatrix, 32
- assembleConductivityMatrix, 32
- assembleExternalForces, 32
- assembleExternalHeat, 32
- assembleMassMatrix, 32
- Body, 30
- bondedBodyNodes, 37
- boundaryConnectivity, 37
- boundaryGroups, 37
- calcCapacityMatrix, 32
- calcConductivityMatrix, 32
- calcExternalForces, 33
- calcExternalHeat, 33
- calcMassMatrix, 33
- cells, 37
- computeEnergy, 37
- getBoundaryFirstNode, 33
- getBoundaryNextNode, 33
- getBoundarySize, 33
- getCellLastNumber, 34
- getLastNode, 34
- getNode, 34
- getNodes, 34
- getNodesSize, 34
- getType, 34
- initialize, 34
- isThermal, 37
- lastNode, 37
- linearBoundary, 37
- loadThermalBody, 37
- nodes, 37
- outputStep, 35
- outputToFile, 36
- setLoadThermal, 36
- setLoadThermalInBoundaryGroup, 36
- setMechanical, 36
- setOutput, 36
- setTemperature, 36
- temperature, 37
- title, 37
- translate, 36
- writeBodyInfo, 36
- writeBoundaryConnectivity, 36
- writeBoundaryNodes, 37
- mknix::BoundaryGroup, 38
  - ~BoundaryGroup, 39
  - addCell, 39
  - addNode, 39
  - assembleExternalHeat, 39
  - BoundaryGroup, 39
  - calcExternalHeat, 39
  - cells, 40
  - initialize, 39
  - loadThermalBoundaryGroup, 40
  - nodes, 40
  - setLoadThermal, 39
- mknix::Cell, 40
  - ~Cell, 42
  - alpha, 44
  - assembleCapacityGaussPoints, 42
  - assembleConductivityGaussPoints, 42
  - assembleFextGaussPoints, 42
  - assembleFintGaussPoints, 42
  - assembleKGaussPoints, 42
  - assembleMGaussPoints, 42
  - assembleNLRGaussPoints, 42
  - assembleQextGaussPoints, 43
  - assembleRGaussPoints, 43
  - bodyPoints, 44
  - calcElasticEGaussPoints, 43
  - calcKineticEGaussPoints, 43
  - calcPotentialEGaussPoints, 43
  - Cell, 42
  - computeCapacityGaussPoints, 43
  - computeConductivityGaussPoints, 43
  - computeFextGaussPoints, 43
  - computeFintGaussPoints, 43
  - computeKGaussPoints, 43
  - computeMGaussPoints, 43
  - computeNLFintGaussPoints, 43
  - computeNLKGaussPoints, 43
  - computeQextGaussPoints, 43
  - computeShapeFunctions, 43
  - dc, 44
  - formulation, 44
  - gPoints, 44
  - gPoints\_MC, 44
  - gnuplotOut, 44
  - gnuplotOutStress, 44
  - initialize, 44
  - jacobian, 44
  - mat, 44
  - nGPoints, 44
  - outputConnectivityToFile, 44
- mknix::CellBoundary, 45
  - ~CellBoundary, 46
  - alpha, 47
  - assembleQextGaussPoints, 46
  - bodyPoints, 47
  - CellBoundary, 46
  - computeQextGaussPoints, 46
  - computeShapeFunctions, 46
  - dc, 47
  - formulation, 47
  - gPoints, 47
  - initialize, 46
  - jacobian, 47
  - nGPoints, 47
- mknix::CellBoundaryLinear, 47
  - ~CellBoundaryLinear, 49
  - CellBoundaryLinear, 49

- computeShapeFunctions, 50
- createGaussPoints, 50
- initialize, 50
- points, 51
- mknix::CellRect, 51
  - ~CellRect, 52
  - CellRect, 52
  - gnuplotOut, 52
  - initialize, 52
- mknix::CellTetrahedron, 53
  - ~CellTetrahedron, 54
  - CellTetrahedron, 54
  - createGaussPoints, 55
  - gnuplotOut, 55
  - points, 55
- mknix::CellTriang, 55
  - ~CellTriang, 58
  - CellTriang, 57
  - createGaussPoints, 58
  - gnuplotOut, 58
  - points, 58
- mknix::CompBar, 59
  - ~CompBar, 59
  - addToRender, 59
  - CompBar, 59
  - removeFromRender, 59
  - updatePoints, 59
- mknix::Constraint, 59
  - ~Constraint, 61
  - alpha, 64
  - assembleInternalForces, 61
  - assembleTangentMatrix, 62
  - calcInternalForces, 62
  - calcPhi, 62
  - calcPhiq, 63
  - calcPhiqq, 63
  - calcTangentMatrix, 63
  - checkAugmented, 64
  - clearAugmented, 64
  - Constraint, 61
  - dim, 64
  - getNode, 64
  - internalForces, 64
  - internalForcesOutput, 64
  - lambda, 65
  - method, 65
  - nodes, 65
  - outputStep, 64
  - outputToFile, 64
  - phi, 65
  - phi\_q, 65
  - phi\_qq, 65
  - setTitle, 64
  - stiffnessMatrix, 65
  - title, 65
  - writeJointInfo, 64
- mknix::ConstraintClearance, 65
  - ~ConstraintClearance, 66
- calcPhi, 67
- calcPhiq, 67
- calcPhiqq, 67
- ConstraintClearance, 66
- getInternalForces, 67
- getStiffnessMatrix, 67
- rh, 67
- rt, 67
- mknix::ConstraintContact, 67
  - ~ConstraintContact, 69
  - calcPhi, 69
  - calcPhiq, 69
  - calcPhiqq, 69
  - ConstraintContact, 69
  - getGap, 69
  - getInternalForces, 69
  - getStiffnessMatrix, 69
  - normal, 69
  - rh, 70
  - rt, 70
- mknix::ConstraintDistance, 70
  - ~ConstraintDistance, 71
  - calcPhi, 72
  - calcPhiq, 72
  - calcPhiqq, 72
  - calcRo, 72
  - ConstraintDistance, 71, 72
  - getInternalForces, 72
  - getStiffnessMatrix, 73
  - ro, 73
  - rt, 73
  - setLenght, 73
- mknix::ConstraintFixedAxis, 73
  - ~ConstraintFixedAxis, 75
  - axisName, 75
  - calcPhi, 75
  - calcPhiq, 75
  - calcPhiqq, 75
  - ConstraintFixedAxis, 74
  - getInternalForces, 75
  - getStiffnessMatrix, 75
  - ro, 75
  - rt, 75
- mknix::ConstraintFixedCoordinates, 76
  - ~ConstraintFixedCoordinates, 77
  - calcPhi, 77
  - calcPhiq, 77
  - calcPhiqq, 78
  - ConstraintFixedCoordinates, 77
  - getInternalForces, 78
  - getStiffnessMatrix, 78
  - rxo, 78
  - rxt, 78
  - ryo, 78
  - ryt, 78
  - rzo, 78
  - rzt, 78
- mknix::ConstraintThermal, 78

- ~ConstraintThermal, 80
  - assembleInternalForces, 80
  - assembleTangentMatrix, 80
  - ConstraintThermal, 80
- mknix::ConstraintThermalFixed, 80
  - ~ConstraintThermalFixed, 82
  - calcPhi, 82
  - calcPhiq, 82
  - calcPhiqq, 83
  - ConstraintThermalFixed, 82
  - getInternalForces, 83
  - getStiffnessMatrix, 83
  - To, 83
  - Tt, 83
- mknix::Contact, 83
  - ~Contact, 84
  - Contact, 84
  - createDelaunay, 84
  - createDrawingContactObjects, 84
  - createDrawingObjects, 84
  - createPoints, 84
  - createPolys, 84
  - drawObjects, 84
  - updateDelaunay, 84
  - updateLines, 84
  - updatePoints, 84
- mknix::ElemTetrahedron, 84
  - ~ElemTetrahedron, 86
  - computeShapeFunctions, 86
  - createGaussPoints\_MC, 86
  - ElemTetrahedron, 86
  - initialize, 86
- mknix::ElemTriangle, 87
  - ~ElemTriangle, 89
  - computeShapeFunctions, 89
  - createGaussPoints\_MC, 89
  - ElemTriangle, 88
  - initialize, 89
- mknix::FlexBody, 90
  - ~FlexBody, 91
  - addBodyPoint, 91
  - addPoint, 91, 92
  - assembleInternalForces, 92
  - assembleTangentMatrix, 92
  - bodyPoints, 95
  - calcInternalForces, 92
  - calcTangentMatrix, 92
  - computeEnergy, 95
  - computeStress, 95
  - energies, 96
  - FlexBody, 91
  - formulation, 96
  - getBodyPoint, 92
  - getLastBodyPoint, 92
  - getNode, 92
  - getNumberOfPoints, 93
  - initialize, 93
  - outputToFile, 93
  - points, 96
  - setFormulation, 94
  - setOutput, 94
  - setType, 94
  - smoothingMassMatrix, 96
  - stresses, 96
  - writeBodyInfo, 94
  - writeBoundaryConnectivity, 95
  - writeBoundaryNodes, 95
- mknix::FlexFrameGalerkin, 96
  - ~FlexFrameGalerkin, 98
  - assembleExternalForces, 98
  - assembleInternalForces, 98
  - assembleMassMatrix, 98
  - assembleTangentMatrix, 98
  - calcExternalForces, 98
  - calcInternalForces, 99
  - calcMassMatrix, 99
  - calcTangentMatrix, 99
  - FlexFrameGalerkin, 98
  - getType, 99
  - outputStep, 99
  - setType, 101
- mknix::FlexGlobalGalerkin, 101
  - ~FlexGlobalGalerkin, 103
  - assembleExternalForces, 103
  - assembleInternalForces, 103
  - assembleMassMatrix, 103
  - assembleTangentMatrix, 104
  - calcExternalForces, 104
  - calcInternalForces, 104
  - calcMassMatrix, 104
  - calcTangentMatrix, 104
  - FlexGlobalGalerkin, 103
  - getType, 104
  - outputStep, 104, 105
  - setType, 105
- mknix::Force, 106
  - ~Force, 107
  - Force, 107
  - outputToFile, 107
- mknix::GaussPoint, 107
  - ~GaussPoint, 109
  - assembleCij, 110
  - assembleFext, 110
  - assembleFint, 110
  - assembleHij, 110
  - assembleKij, 110
  - assembleMij, 110
  - assembleQext, 110
  - assembleRi, 110
  - B, 113
  - C, 113
  - calcElasticE, 110
  - calcKineticE, 110
  - calcPotentialE, 110
  - computeCij, 110
  - computeFext, 111

- computeFint, [111](#)
- computeHij, [111](#)
- computeKij, [111](#)
- computeMij, [111](#)
- computeNLFint, [111](#)
- computeNLKij, [111](#)
- computeNLStress, [111](#)
- computeQext, [111](#)
- computeStress, [112](#)
- fext, [113](#)
- fillFEmatrices, [112](#)
- fint, [113](#)
- GaussPoint, [109](#)
- gnuplotOutStress, [112](#)
- H, [113](#)
- K, [113](#)
- M, [113](#)
- mat, [113](#)
- num, [113](#)
- Qext, [113](#)
- r, [113](#)
- shapeFunSolve, [112](#)
- stressPoint, [114](#)
- tension, [114](#)
- weight, [114](#)
- mknix::GaussPoint2D, [114](#)
  - ~GaussPoint2D, [116](#)
  - assembleFext, [116](#)
  - assembleFint, [116](#)
  - assembleKij, [116](#)
  - assembleMij, [116](#)
  - assembleRi, [116](#)
  - calcElasticE, [117](#)
  - calcKineticE, [117](#)
  - calcPotentialE, [117](#)
  - computeFext, [118](#)
  - computeFint, [118](#)
  - computeKij, [118](#)
  - computeMij, [119](#)
  - computeNLFint, [119](#)
  - computeNLKij, [120](#)
  - computeNLStress, [120](#)
  - computeStress, [121](#)
  - fillFEmatrices, [121](#)
  - GaussPoint2D, [116](#)
  - shapeFunSolve, [122](#)
- mknix::GaussPoint3D, [122](#)
  - ~GaussPoint3D, [124](#)
  - assembleFext, [124](#)
  - assembleFint, [124](#)
  - assembleKij, [124](#)
  - assembleMij, [124](#)
  - assembleRi, [124](#)
  - calcElasticE, [125](#)
  - calcKineticE, [125](#)
  - calcPotentialE, [125](#)
  - computeFext, [126](#)
  - computeFint, [126](#)
  - computeKij, [126](#)
  - computeMij, [127](#)
  - computeNLFint, [127](#)
  - computeNLKij, [128](#)
  - computeNLStress, [128](#)
  - computeStress, [129](#)
  - fillFEmatrices, [129](#)
  - GaussPoint3D, [124](#)
  - shapeFunSolve, [130](#)
- mknix::GaussPointBoundary, [130](#)
  - ~GaussPointBoundary, [132](#)
  - assembleQext, [132](#)
  - computeQext, [132](#)
  - GaussPointBoundary, [132](#)
  - initializeMatVecs, [132](#)
  - num, [133](#)
  - Qext, [133](#)
  - shapeFunSolve, [133](#)
  - weight, [133](#)
- mknix::Load, [133](#)
  - ~Load, [135](#)
  - assembleExternalForces, [135](#)
  - externalForces, [135](#)
  - externalHeat, [135](#)
  - Load, [135](#)
  - nodes, [135](#)
  - outputToFile, [135](#)
- mknix::LoadThermal, [135](#)
  - ~LoadThermal, [136](#)
  - assembleExternalHeat, [136](#)
  - externalHeat, [137](#)
  - getMaxTemp, [136](#)
  - insertNodesXCoordinates, [136](#)
  - LoadThermal, [136](#)
  - nodes, [137](#)
  - outputToFile, [136](#)
  - updateLoad, [136](#)
- mknix::LoadThermalBody, [137](#)
  - ~LoadThermalBody, [137](#)
  - getLoadThermalBody, [138](#)
  - LoadThermalBody, [137](#)
  - srin, [138](#)
- mknix::LoadThermalBoundary1D, [138](#)
  - ~LoadThermalBoundary1D, [139](#)
  - getLoadThermalBoundary1D, [139](#)
  - loadFile, [140](#)
  - LoadThermalBoundary1D, [139](#)
  - loadTimeFile, [140](#)
  - loadmap, [140](#)
  - scaleLoad, [140](#)
  - timemap, [140](#)
- mknix::Material, [140](#)
  - ~Material, [141](#)
  - addThermalCapacity, [141](#)
  - addThermalConductivity, [141](#)
  - computeC, [141](#)
  - computeD, [141](#)
  - computeEnergy, [142](#)

- computeS, [142](#), [143](#)
- getC, [143](#)
- getCapacity, [143](#)
- getCsym, [144](#)
- getD, [144](#)
- getDensity, [144](#)
- getE, [145](#)
- getKappa, [145](#)
- getMu, [146](#)
- Material, [141](#)
- setMechanicalProps, [146](#)
- setThermalProps, [146](#)
- mknix::Motion, [147](#)
  - ~Motion, [148](#)
  - Motion, [148](#)
  - setNode, [148](#)
  - setTimeUx, [148](#)
  - setTimeUy, [148](#)
  - setTimeUz, [148](#)
  - update, [148](#)
- mknix::Node, [149](#)
  - ~Node, [150](#)
  - addWeight, [150](#)
  - getConf, [150](#)
  - getShapeFunValue, [152](#)
  - getSupportNodeNumber, [152](#)
  - getSupportSize, [152](#)
  - getTemp, [153](#)
  - getThermalNumber, [153](#)
  - getU, [153](#)
  - getUx, [154](#)
  - getUy, [154](#)
  - getUz, [154](#)
  - getWeight, [155](#)
  - getqt, [151](#)
  - getqx, [152](#)
  - Node, [150](#)
  - setNumber, [155](#)
  - setThermalNumber, [155](#)
  - setUx, [155](#)
  - setUy, [155](#)
  - setUz, [156](#)
  - setX, [156](#)
  - setY, [156](#)
  - setZ, [156](#)
  - setqt, [155](#)
  - setqx, [155](#)
- mknix::Point, [156](#)
  - ~Point, [159](#)
  - addSupportNode, [159](#)
  - alpha, [167](#)
  - dc, [167](#)
  - dim, [167](#)
  - distance, [159](#)
  - findSupportNodes, [160](#)
  - GaussPoint, [166](#)
  - getConf, [160](#)
  - getDim, [161](#)
  - getNumber, [161](#)
  - getShapeFunType, [161](#)
  - getSupportNodeNumber, [161](#)
  - getSupportNodes, [162](#)
  - getSupportSize, [162](#)
  - getTemp, [162](#)
  - getX, [162](#)
  - getY, [163](#)
  - getZ, [164](#)
  - gnuplotOut, [164](#)
  - jacobian, [167](#)
  - num, [167](#)
  - Point, [158](#), [159](#)
  - setAlpha, [165](#)
  - setDc, [165](#)
  - setJacobian, [165](#)
  - setShapeFunType, [165](#)
  - shapeFun, [167](#)
  - shapeFunSolve, [165](#)
  - shapeFunType, [167](#)
  - ShapeFunction, [166](#)
  - ShapeFunctionLinear, [166](#)
  - ShapeFunctionLinearX, [166](#)
  - ShapeFunctionMLS, [166](#)
  - ShapeFunctionRBF, [166](#)
  - ShapeFunctionTetrahedron, [166](#)
  - ShapeFunctionTriangle, [166](#)
  - supportNodes, [167](#)
  - supportNodesSize, [167](#)
  - X, [167](#)
  - Y, [167](#)
  - Z, [167](#)
- mknix::Radiation, [168](#)
  - ~Radiation, [169](#)
  - addVoxel, [169](#)
  - outputToFile, [169](#)
  - Radiation, [169](#)
- mknix::Reader, [169](#)
  - ~Reader, [170](#)
  - inputFromFile, [170](#)
  - Reader, [169](#)
- mknix::ReaderConstraints, [170](#)
  - ~ReaderConstraints, [171](#)
  - readConstraints, [171](#)
  - ReaderConstraints, [171](#)
- mknix::ReaderFlex, [171](#)
  - ~ReaderFlex, [172](#)
  - readFlexBodies, [172](#)
  - ReaderFlex, [172](#)
- mknix::ReaderRigid, [172](#)
  - ~ReaderRigid, [173](#)
  - readRigidBodies, [173](#)
  - ReaderRigid, [173](#)
- mknix::RigidBar, [173](#)
  - ~RigidBar, [175](#)
  - addNode, [175](#)
  - calcExternalForces, [176](#)
  - calcMassMatrix, [176](#)



- getType, 176
  - RigidBody, 175
  - setInertia, 176
  - setPosition, 177
  - writeBoundaryConnectivity, 177
  - writeBoundaryNodes, 177
- mknix::RigidBody, 177
  - ~RigidBody, 179
  - assembleExternalForces, 179
  - assembleMassMatrix, 179
  - computeEnergy, 182
  - densityFactor, 182
  - dim, 182
  - domainConf, 182
  - energy, 182
  - externalForces, 182
  - frameNodes, 182
  - getNode, 180
  - localMassMatrix, 182
  - mass, 182
  - outputStep, 180
  - outputToFile, 180
  - RigidBody, 179
  - setDensityFactor, 181
  - setInertia, 181
  - setMass, 181
  - setOutput, 181
  - setPosition, 181
  - writeBodyInfo, 181
  - writeBoundaryConnectivity, 182
  - writeBoundaryNodes, 182
- mknix::RigidBody2D, 182
  - ~RigidBody2D, 184
  - addNode, 184
  - calcExternalForces, 184
  - calcMassMatrix, 185
  - getType, 185
  - RigidBody2D, 184
  - setInertia, 185
  - setPosition, 185
- mknix::RigidBody3D, 186
  - ~RigidBody3D, 187
  - addNode, 187
  - calcExternalForces, 187
  - calcMassMatrix, 188
  - getType, 188
  - RigidBody3D, 187
  - setInertia, 188
  - setPosition, 188
- mknix::RigidBodyMassPoint, 188
  - ~RigidBodyMassPoint, 190
  - calcExternalForces, 190
  - calcMassMatrix, 190
  - getType, 190
  - RigidBodyMassPoint, 190
  - setInertia, 190
  - setPosition, 190
- mknix::ShapeFunction, 191
  - ~ShapeFunction, 193
  - calc, 193
  - dim, 195
  - getPhi, 193
  - gnuplotOut, 194
  - gp, 195
  - outputValues, 195
  - phi, 195
  - setPhi, 195
  - ShapeFunction, 192, 193
- mknix::ShapeFunctionLinear, 195
  - ~ShapeFunctionLinear, 197
  - calc, 197
  - ShapeFunctionLinear, 197
- mknix::ShapeFunctionLinearX, 197
  - ~ShapeFunctionLinearX, 199
  - calc, 199
  - ShapeFunctionLinearX, 199
- mknix::ShapeFunctionMLS, 199
  - ~ShapeFunctionMLS, 201
  - calc, 201
  - ShapeFunctionMLS, 201
- mknix::ShapeFunctionRBF, 202
  - ~ShapeFunctionRBF, 203
  - calc, 203
  - ShapeFunctionRBF, 203
- mknix::ShapeFunctionTetrahedron, 203
  - ~ShapeFunctionTetrahedron, 205
  - calc, 205
  - compute\_abcd, 205
  - ShapeFunctionTetrahedron, 205
- mknix::ShapeFunctionTriangle, 205
  - ~ShapeFunctionTriangle, 207
  - calc, 207
  - ShapeFunctionTriangle, 207
- mknix::ShapeFunctionTriangleSigned, 207
  - ~ShapeFunctionTriangleSigned, 208
  - calc, 209
  - ShapeFunctionTriangleSigned, 208
- mknix::Simulation, 209
  - ~Simulation, 211
  - Contact, 227
  - dynamicAcceleration, 211
  - dynamicConvergence, 211
  - dynamicResidue, 212
  - dynamicTangent, 212
  - dynamicThermalConvergence, 213
  - dynamicThermalConvergenceInThermomechanical, 213
  - dynamicThermalEvaluation, 213
  - dynamicThermalResidue, 214
  - dynamicThermalTangent, 214
  - endSimulation, 214
  - explicitAcceleration, 214
  - explicitThermalEvaluation, 215
  - getAlpha, 215
  - getConstraintMethod, 216
  - getDim, 216

- getGravity, [217](#)
- getInterfaceNodesCoords, [218](#)
- getSmoothingType, [218](#)
- getSparsePattern, [219](#)
- getTime, [220](#)
- init, [220](#)
- inputFromFile, [220](#)
- operator=, [220](#)
- Reader, [227](#)
- ReaderConstraints, [227](#)
- ReaderFlex, [227](#)
- ReaderRigid, [227](#)
- run, [220](#)
- runMechanicalAnalysis, [221](#)
- runThermalAnalysis, [221](#)
- setInitialTemperatures, [222](#)
- setOutputFilesDetail, [222](#)
- Simulation, [211](#)
- solveStep, [222](#), [223](#)
- staticConvergence, [223](#)
- staticResidue, [224](#)
- staticTangent, [224](#)
- staticThermalConvergence, [225](#)
- staticThermalResidue, [225](#)
- staticThermalTangent, [225](#)
- stepTriggered, [226](#)
- SystemChain, [227](#)
- writeConfStep, [226](#)
- writeSystem, [227](#)
- mnkix::System, [228](#)
  - ~System, [230](#)
  - assembleCapacityMatrix, [230](#)
  - assembleConductivityMatrix, [230](#)
  - assembleConstraintForces, [230](#)
  - assembleExternalForces, [230](#)
  - assembleExternalHeat, [230](#)
  - assembleInternalForces, [230](#)
  - assembleInternalHeat, [230](#)
  - assembleMassMatrix, [230](#)
  - assembleTangentMatrix, [230](#)
  - assembleThermalTangentMatrix, [230](#)
  - calcCapacityMatrix, [230](#)
  - calcConductivityMatrix, [230](#)
  - calcExternalForces, [231](#)
  - calcExternalHeat, [231](#)
  - calcInternalForces, [231](#)
  - calcInternalHeat, [231](#)
  - calcMassMatrix, [231](#)
  - calcTangentMatrix, [231](#)
  - calcThermalTangentMatrix, [231](#)
  - checkAugmented, [231](#)
  - clearAugmented, [231](#)
  - constraints, [233](#)
  - constraintsThermal, [233](#)
  - Contact, [233](#)
  - flexBodies, [233](#)
  - getNode, [231](#)
  - getOutputSignalThermal, [231](#)
  - getThermalNodes, [231](#)
  - getTitle, [231](#)
  - groundNodes, [233](#)
  - initFlexBodies, [232](#)
  - loads, [233](#)
  - loadsThermal, [234](#)
  - motions, [234](#)
  - outputMaxInterfaceTemp, [234](#)
  - outputSignalThermal, [234](#)
  - outputStep, [232](#)
  - outputToFile, [232](#)
  - Reader, [233](#)
  - ReaderConstraints, [233](#)
  - ReaderFlex, [233](#)
  - ReaderRigid, [233](#)
  - rigidBodies, [234](#)
  - setMechanical, [232](#)
  - subSystems, [234](#)
  - System, [230](#)
  - thermalBodies, [234](#)
  - title, [234](#)
  - update, [232](#)
  - updateThermalLoads, [232](#)
  - writeBoundaryConnectivity, [232](#)
  - writeBoundaryNodes, [233](#)
  - writeFlexBodies, [233](#)
  - writeJoints, [233](#)
  - writeRigidBodies, [233](#)
- mnkix::SystemChain, [234](#)
  - ~SystemChain, [236](#)
  - addTimeLenght, [236](#)
  - getNode, [236](#)
  - populate, [236](#)
  - setInterfaceNodeA, [236](#)
  - setInterfaceNodeB, [236](#)
  - setMass, [236](#), [237](#)
  - setProperties, [237](#)
  - setTimeLengths, [237](#)
  - SystemChain, [236](#)
  - update, [237](#)
- mnkix::ThermalBody, [237](#)
  - ~ThermalBody, [239](#)
  - addCell, [239](#)
  - addNode, [239](#)
  - assembleCapacityMatrix, [239](#)
  - assembleConductivityMatrix, [239](#)
  - assembleExternalHeat, [240](#)
  - calcCapacityMatrix, [240](#)
  - calcConductivityMatrix, [240](#)
  - calcExternalHeat, [240](#)
  - cells, [242](#)
  - computeEnergy, [242](#)
  - getLastNode, [240](#)
  - getNode, [240](#)
  - initialize, [241](#)
  - loadThermalBody, [242](#)
  - nodes, [242](#)
  - outputStep, [241](#)

- outputToFile, 241
- setLoadThermal, 241
- setOutput, 242
- temperature, 242
- ThermalBody, 239
- title, 242
- Motion
  - mknix::Motion, 148
- motion.cpp, 323
- motion.h, 324
- motions
  - mknix::System, 234
- nGPoints
  - mknix::Cell, 44
  - mknix::CellBoundary, 47
- nextStep
  - mknix::Analysis, 15
  - mknix::AnalysisThermalDynamic, 23
- Node
  - mknix::Node, 150
- node.cpp, 325
- node.h, 326
- nodes
  - mknix::Body, 37
  - mknix::BoundaryGroup, 40
  - mknix::Constraint, 65
  - mknix::Load, 135
  - mknix::LoadThermal, 137
  - mknix::ThermalBody, 242
- normal
  - mknix::ConstraintContact, 69
- num
  - mknix::GaussPoint, 113
  - mknix::GaussPointBoundary, 133
  - mknix::Point, 167
- operator=
  - mknix::Simulation, 220
- outputConnectivityToFile
  - mknix::Cell, 44
- outputMaxInterfaceTemp
  - mknix::System, 234
- outputSignalThermal
  - mknix::System, 234
- outputStep
  - mknix::Body, 35
  - mknix::Constraint, 64
  - mknix::FlexFrameGalerkin, 99
  - mknix::FlexGlobalGalerkin, 104, 105
  - mknix::RigidBody, 180
  - mknix::System, 232
  - mknix::ThermalBody, 241
- outputToFile
  - mknix::Body, 36
  - mknix::Constraint, 64
  - mknix::FlexBody, 93
  - mknix::Force, 107
  - mknix::Load, 135
  - mknix::LoadThermal, 136
  - mknix::Radiation, 169
  - mknix::RigidBody, 180
  - mknix::System, 232
  - mknix::ThermalBody, 241
- outputValues
  - mknix::ShapeFunction, 195
- phi
  - mknix::Constraint, 65
  - mknix::ShapeFunction, 195
- phi\_q
  - mknix::Constraint, 65
- phi\_qq
  - mknix::Constraint, 65
- Point
  - mknix::Point, 158, 159
- point.cpp, 327
- point.h, 328
- points
  - mknix::CellBoundaryLinear, 51
  - mknix::CellTetrahedron, 55
  - mknix::CellTriang, 58
  - mknix::FlexBody, 96
- populate
  - mknix::SystemChain, 236
- Qext
  - mknix::GaussPoint, 113
  - mknix::GaussPointBoundary, 133
- r
  - mknix::GaussPoint, 113
- Radiation
  - mknix::Radiation, 169
- readConstraints
  - mknix::ReaderConstraints, 171
- readFlexBodies
  - mknix::ReaderFlex, 172
- readRigidBodies
  - mknix::ReaderRigid, 173
- Reader
  - mknix::Reader, 169
  - mknix::Simulation, 227
  - mknix::System, 233
- reader.cpp, 329
- reader.h, 329
- ReaderConstraints
  - mknix::ReaderConstraints, 171
  - mknix::Simulation, 227
  - mknix::System, 233
- ReaderFlex
  - mknix::ReaderFlex, 172
  - mknix::Simulation, 227
  - mknix::System, 233
- ReaderRigid
  - mknix::ReaderRigid, 173
  - mknix::Simulation, 227
  - mknix::System, 233

readerconstraints.cpp, 330  
 readerconstraints.h, 331  
 readerflex.cpp, 332  
 readerflex.h, 333  
 readerrigid.cpp, 333  
 readerrigid.h, 334  
 removeFromRender  
     mknix::CompBar, 59  
 rh  
     mknix::ConstraintClearance, 67  
     mknix::ConstraintContact, 70  
 RigidBar  
     mknix::RigidBar, 175  
 rigidBodies  
     mknix::System, 234  
 RigidBody  
     mknix::RigidBody, 179  
 RigidBody2D  
     mknix::RigidBody2D, 184  
 RigidBody3D  
     mknix::RigidBody3D, 187  
 RigidBodyMassPoint  
     mknix::RigidBodyMassPoint, 190  
 ro  
     mknix::ConstraintDistance, 73  
     mknix::ConstraintFixedAxis, 75  
 rt  
     mknix::ConstraintClearance, 67  
     mknix::ConstraintContact, 70  
     mknix::ConstraintDistance, 73  
     mknix::ConstraintFixedAxis, 75  
 run  
     mknix::Simulation, 220  
 runMechanicalAnalysis  
     mknix::Simulation, 221  
 runThermalAnalysis  
     mknix::Simulation, 221  
 rxo  
     mknix::ConstraintFixedCoordinates, 78  
 rxt  
     mknix::ConstraintFixedCoordinates, 78  
 ryo  
     mknix::ConstraintFixedCoordinates, 78  
 ryt  
     mknix::ConstraintFixedCoordinates, 78  
 rzo  
     mknix::ConstraintFixedCoordinates, 78  
 rzt  
     mknix::ConstraintFixedCoordinates, 78  
 scaleLoad  
     mknix::LoadThermalBoundary1D, 140  
 setAlpha  
     mknix::Point, 165  
 setDc  
     mknix::Point, 165  
 setDensityFactor  
     mknix::RigidBody, 181  
 setEpsilon  
     mknix::Analysis, 15  
 setFormulation  
     mknix::FlexBody, 94  
 setInertia  
     mknix::RigidBar, 176  
     mknix::RigidBody, 181  
     mknix::RigidBody2D, 185  
     mknix::RigidBody3D, 188  
     mknix::RigidBodyMassPoint, 190  
 setInitialTemperatures  
     mknix::Simulation, 222  
 setInterfaceNodeA  
     mknix::SystemChain, 236  
 setInterfaceNodeB  
     mknix::SystemChain, 236  
 setJacobian  
     mknix::Point, 165  
 setLenght  
     mknix::ConstraintDistance, 73  
 setLoadThermal  
     mknix::Body, 36  
     mknix::BoundaryGroup, 39  
     mknix::ThermalBody, 241  
 setLoadThermalInBoundaryGroup  
     mknix::Body, 36  
 setMass  
     mknix::RigidBody, 181  
     mknix::SystemChain, 236, 237  
 setMechanical  
     mknix::Body, 36  
     mknix::System, 232  
 setMechanicalProps  
     mknix::Material, 146  
 setNode  
     mknix::Motion, 148  
 setNumber  
     mknix::Node, 155  
 setOutput  
     mknix::Body, 36  
     mknix::FlexBody, 94  
     mknix::RigidBody, 181  
     mknix::ThermalBody, 242  
 setOutputFilesDetail  
     mknix::Simulation, 222  
 setPhi  
     mknix::ShapeFunction, 195  
 setPosition  
     mknix::RigidBar, 177  
     mknix::RigidBody, 181  
     mknix::RigidBody2D, 185  
     mknix::RigidBody3D, 188  
     mknix::RigidBodyMassPoint, 190  
 setProperties  
     mknix::SystemChain, 237  
 setShapeFunType  
     mknix::Point, 165  
 setSignal  
     mknix, 13

- setTemperature
  - mknix::Body, 36
- setThermalNumber
  - mknix::Node, 155
- setThermalProps
  - mknix::Material, 146
- setTimeLengths
  - mknix::SystemChain, 237
- setTimeUx
  - mknix::Motion, 148
- setTimeUy
  - mknix::Motion, 148
- setTimeUz
  - mknix::Motion, 148
- setTitle
  - mknix::Constraint, 64
- setType
  - mknix::FlexBody, 94
  - mknix::FlexFrameGalerkin, 101
  - mknix::FlexGlobalGalerkin, 105
- setUx
  - mknix::Node, 155
- setUy
  - mknix::Node, 155
- setUz
  - mknix::Node, 156
- setX
  - mknix::Node, 156
- setY
  - mknix::Node, 156
- setZ
  - mknix::Node, 156
- setqt
  - mknix::Node, 155
- setqx
  - mknix::Node, 155
- shapeFun
  - mknix::Point, 167
- shapeFunSolve
  - mknix::GaussPoint, 112
  - mknix::GaussPoint2D, 122
  - mknix::GaussPoint3D, 130
  - mknix::GaussPointBoundary, 133
  - mknix::Point, 165
- shapeFunType
  - mknix::Point, 167
- ShapeFunction
  - mknix::Point, 166
  - mknix::ShapeFunction, 192, 193
- ShapeFunctionLinear
  - mknix::Point, 166
  - mknix::ShapeFunctionLinear, 197
- ShapeFunctionLinearX
  - mknix::Point, 166
  - mknix::ShapeFunctionLinearX, 199
- ShapeFunctionMLS
  - mknix::Point, 166
  - mknix::ShapeFunctionMLS, 201
- ShapeFunctionRBF
  - mknix::Point, 166
  - mknix::ShapeFunctionRBF, 203
- ShapeFunctionTetrahedron
  - mknix::Point, 166
  - mknix::ShapeFunctionTetrahedron, 205
- ShapeFunctionTriangle
  - mknix::Point, 166
  - mknix::ShapeFunctionTriangle, 207
- ShapeFunctionTriangleSigned
  - mknix::ShapeFunctionTriangleSigned, 208
- shapefunction.cpp, 335
- shapefunction.h, 336
- shapefunctionMLS.cpp, 340
- shapefunctionMLS.h, 340
- shapefunctionMLS2D.cpp, 342
- shapefunctionMLS2D.h, 343
- shapefunctionMLS3D.cpp, 343
- shapefunctionMLS3D.h, 344
- shapefunctionRBF.cpp, 345
- shapefunctionRBF.h, 346
- shapefunctionlinear-x.cpp, 336
- shapefunctionlinear-x.h, 337
- shapefunctionlinear.cpp, 338
- shapefunctionlinear.h, 339
- shapefunctiontetrahedron.cpp, 347
- shapefunctiontetrahedron.h, 348
- shapefunctiontriangle.cpp, 348
- shapefunctiontriangle.h, 349
- shapefunctiontriangle3D.cpp, 350
- shapefunctiontriangle3D.h, 351
- Simulation
  - mknix::Simulation, 211
- simulation.cpp, 352
- simulation.h, 352
- smoothingMassMatrix
  - mknix::FlexBody, 96
- solve
  - mknix::Analysis, 16
  - mknix::AnalysisDynamic, 18
  - mknix::AnalysisStatic, 21
  - mknix::AnalysisThermalDynamic, 23
  - mknix::AnalysisThermalStatic, 25
  - mknix::AnalysisThermoMechanicalDynamic, 27
- solveStep
  - mknix::Simulation, 222, 223
- srim
  - mknix::LoadThermalBody, 138
- staticConvergence
  - mknix::Simulation, 223
- staticResidue
  - mknix::Simulation, 224
- staticTangent
  - mknix::Simulation, 224
- staticThermalConvergence
  - mknix::Simulation, 225
- staticThermalResidue
  - mknix::Simulation, 225

- staticThermalTangent
  - mknix::Simulation, 225
- stepTriggered
  - mknix::Simulation, 226
- stiffnessMatrix
  - mknix::Constraint, 65
- stressPoint
  - mknix::GaussPoint, 114
- stresses
  - mknix::FlexBody, 96
- subSystems
  - mknix::System, 234
- supportNodes
  - mknix::Point, 167
- supportNodesSize
  - mknix::Point, 167
- System
  - mknix::System, 230
- system.cpp, 353
- system.h, 354
- SystemChain
  - mknix::Simulation, 227
  - mknix::SystemChain, 236
- systemchain.cpp, 355
- systemchain.h, 355
- systemchain2.cpp, 356
- systemchain2.h, 357
- temperature
  - mknix::Body, 37
  - mknix::ThermalBody, 242
- tension
  - mknix::GaussPoint, 114
- theSimulation
  - mknix::Analysis, 16
- thermalBodies
  - mknix::System, 234
- ThermalBody
  - mknix::ThermalBody, 239
- timemap
  - mknix::LoadThermalBoundary1D, 140
- title
  - mknix::Body, 37
  - mknix::Constraint, 65
  - mknix::System, 234
  - mknix::ThermalBody, 242
- To
  - mknix::ConstraintThermalFixed, 83
- translate
  - mknix::Body, 36
- Tt
  - mknix::ConstraintThermalFixed, 83
- type
  - mknix::Analysis, 16
  - mknix::AnalysisDynamic, 19
  - mknix::AnalysisStatic, 21
  - mknix::AnalysisThermalDynamic, 23
  - mknix::AnalysisThermalStatic, 25
  - mknix::AnalysisThermoMechanicalDynamic, 28
- update
  - mknix::Motion, 148
  - mknix::System, 232
  - mknix::SystemChain, 237
- updateDelaunay
  - mknix::Contact, 84
- updateLines
  - mknix::Contact, 84
- updateLoad
  - mknix::LoadThermal, 136
- updatePoints
  - mknix::CompBar, 59
  - mknix::Contact, 84
- updateThermalLoads
  - mknix::System, 232
- weight
  - mknix::GaussPoint, 114
  - mknix::GaussPointBoundary, 133
- writeBodyInfo
  - mknix::Body, 36
  - mknix::FlexBody, 94
  - mknix::RigidBody, 181
- writeBoundaryConnectivity
  - mknix::Body, 36
  - mknix::FlexBody, 95
  - mknix::RigidBar, 177
  - mknix::RigidBody, 182
  - mknix::System, 232
- writeBoundaryNodes
  - mknix::Body, 37
  - mknix::FlexBody, 95
  - mknix::RigidBar, 177
  - mknix::RigidBody, 182
  - mknix::System, 233
- writeConfStep
  - mknix::Simulation, 226
- writeFlexBodies
  - mknix::System, 233
- writeJointInfo
  - mknix::Constraint, 64
- writeJoints
  - mknix::System, 233
- writeRigidBodies
  - mknix::System, 233
- writeSystem
  - mknix::Simulation, 227
- X
  - mknix::Point, 167
- Y
  - mknix::Point, 167
- Z
  - mknix::Point, 167