MkniX

0.9.0

Generated by Doxygen 1.8.9.1

Sat Dec 5 2015 00:24:54

Contents

1 Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

lmx ??

mknix ??

2 Hierarchical Index

2.1 Class Hierarchy

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

- **mknix::Analysis** 13
  - **mknix::AnalysisDynamic** 17
  - **mknix::AnalysisStatic** 19

3 Class Index

3.1 Class List

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

4 File Index

4.1 File List

Here is a list of all files with brief descriptions:

5 Namespace Documentation

5.1 lmx 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

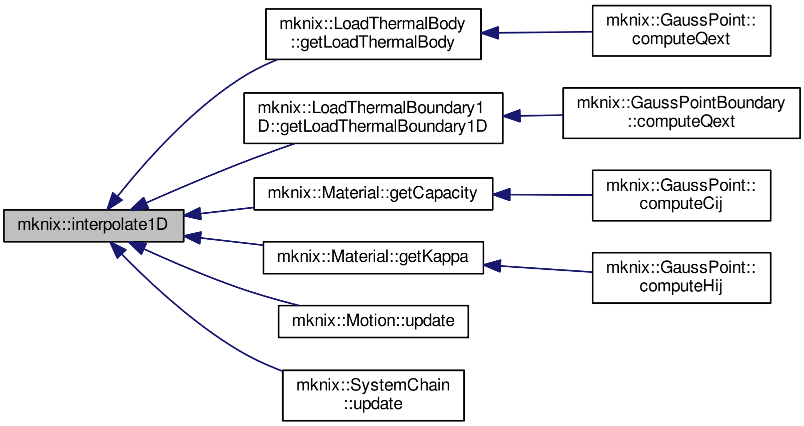
5.2.3.1 std::vector<double> mknix::getSignal (std::string *node*)

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:


mknix::interpolate1D
mknix::LoadThermalBody ::getLoadThermalBody
mknix::LoadThermalBoundary1 D::getLoadThermalBoundary1D
mknix::Material::getCapacity
mknix::Material::getKappa
mknix::Motion::update
mknix::SystemChain ::update
mknix::GaussPoint:: computeQext
mknix::GaussPointBoundary ::computeQext
mknix::GaussPoint:: computeCij
mknix::GaussPoint:: computeHij

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 >)

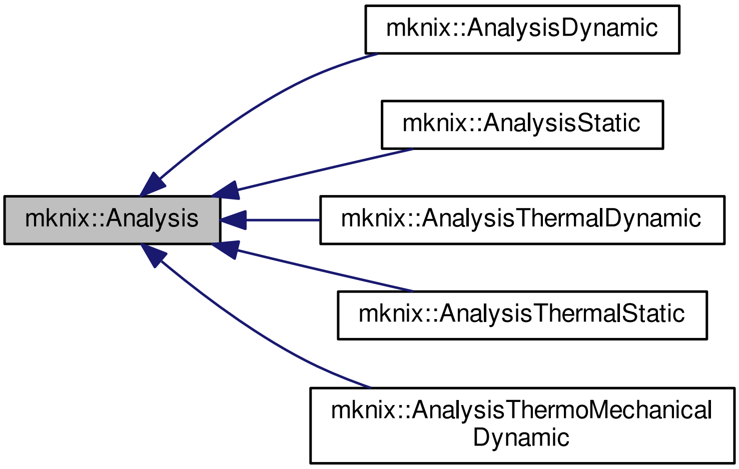
Definition at line 202 of file simulation.cpp.

6 Class Documentation

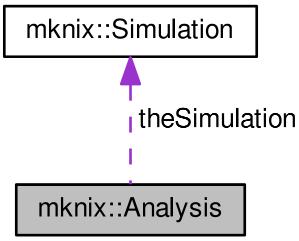
6.1 mknix::Analysis Class Reference

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

Inheritance diagram for mknix::Analysis:


mknix::AnalysisDynamic
mknix::AnalysisStatic
mknix::Analysis
mknix::AnalysisThermalDynamic
mknix::AnalysisThermalStatic
mknix::AnalysisThermoMechanical Dynamic

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 (lmx::Vector< data_type > ∗, lmx::Vector< data_type > ∗=0, lmx::Vector< data_type > ∗=0)=0
- virtual void init (lmx::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 & 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 (lmx::Vector< data_type > ∗) [inline], [virtual]

Reimplemented in mknix::AnalysisThermalDynamic.

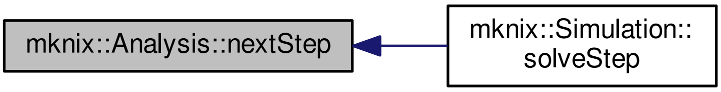
Definition at line 54 of file analysis.h.

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

Reimplemented in mknix::AnalysisThermalDynamic.

Definition at line 55 of file analysis.h.

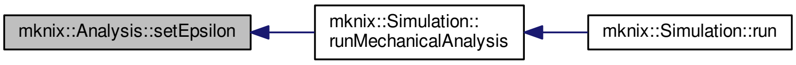
Here is the caller graph for this function:



6.1.3.3 void mknix::Analysis::setEpsilon (double *epsilon_in*) [inline]

Definition at line 45 of file analysis.h.

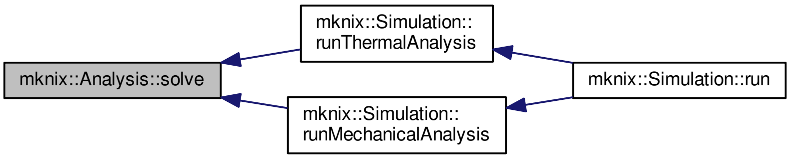
Here is the caller graph for this function:



6.1.3.4 virtual void mknix::Analysis::solve (lmx::Vector< data_type > ∗ , lmx::Vector< data_type > ∗ = 0, lmx::Vector< data_type > ∗ = 0) [pure virtual]

Implemented in mknix::AnalysisThermalDynamic, mknix::AnalysisThermoMechanicalDynamic, mknix::AnalysisDynamic, mknix::AnalysisStatic, and mknix::AnalysisThermalStatic.

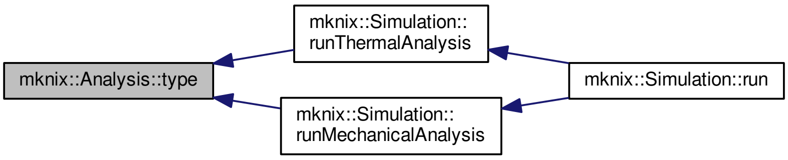
Here is the caller graph for this function:



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.

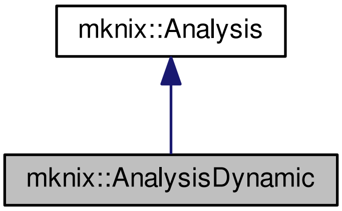
The documentation for this class was generated from the following files:

- analysis.h
- analysis.cpp

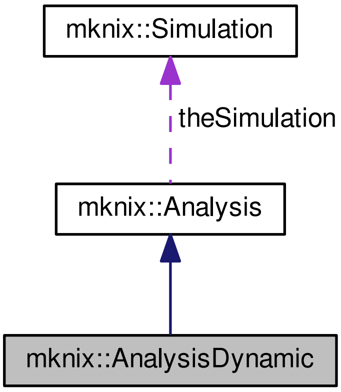
6.2 mknix::AnalysisDynamic Class Reference

```
#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

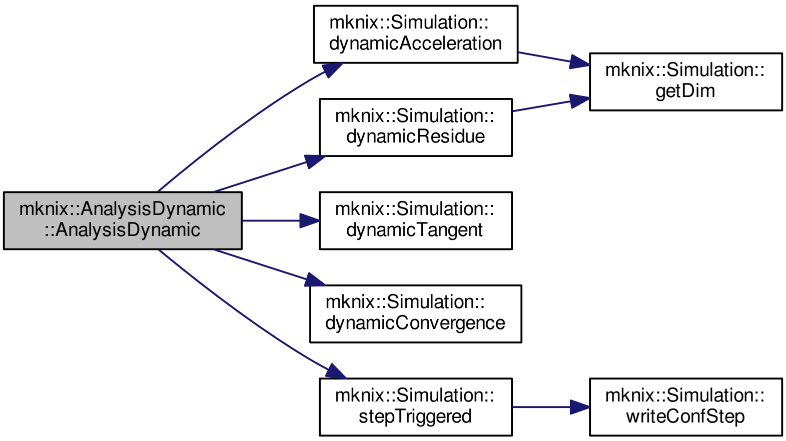
6.2.2.1 mknix::AnalysisDynamic::AnalysisDynamic ( )

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.

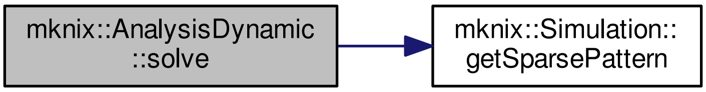
6.2.3 Member Function Documentation

6.2.3.1 void mknix::AnalysisDynamic::solve (lmx::Vector< data_type > ∗ *q_in,* lmx::Vector< data_type > ∗ *qdot_in,* lmx::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:


mknix::AnalysisDynamic ::solve
mknix::Simulation:: getSparsePattern

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

Implements mknix::Analysis.

Definition at line 39 of file analysisdynamic.h.

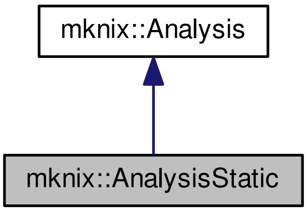
The documentation for this class was generated from the following files:

- analysisdynamic.h
- analysisdynamic.cpp

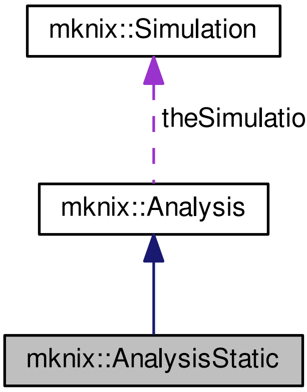
6.3 mknix::AnalysisStatic Class Reference

```
#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

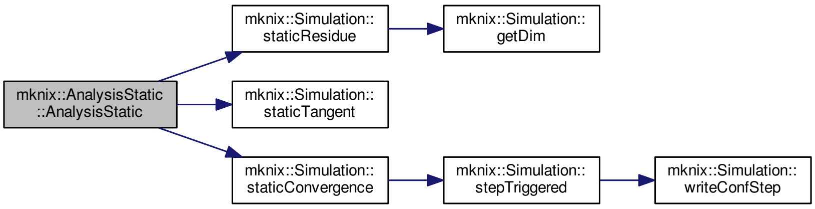
6.3.2.1 mknix::AnalysisStatic::AnalysisStatic ( )

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:


mknix::AnalysisStatic ::AnalysisStatic
mknix::Simulation:: staticResidue
mknix::Simulation:: getDim
mknix::Simulation:: staticTangent
mknix::Simulation:: staticConvergence
mknix::Simulation:: stepTriggered
mknix::Simulation:: writeConfStep

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 (lmx::Vector< data_type > ∗ *q_in*, lmx::Vector< data_type > ∗ *qdot_in* = 0, lmx::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.

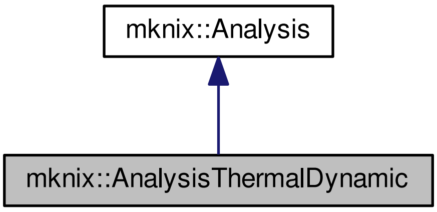
The documentation for this class was generated from the following files:

- analysisstatic.h
- analysisstatic.cpp

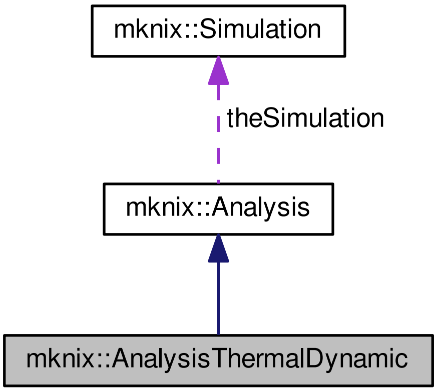
6.4 mknix::AnalysisThermalDynamic Class Reference

```
#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 (lmx::Vector< data_type > ∗)
- void nextStep ()
- void solve (lmx::Vector< data_type > ∗, lmx::Vector< data_type > ∗, lmx::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

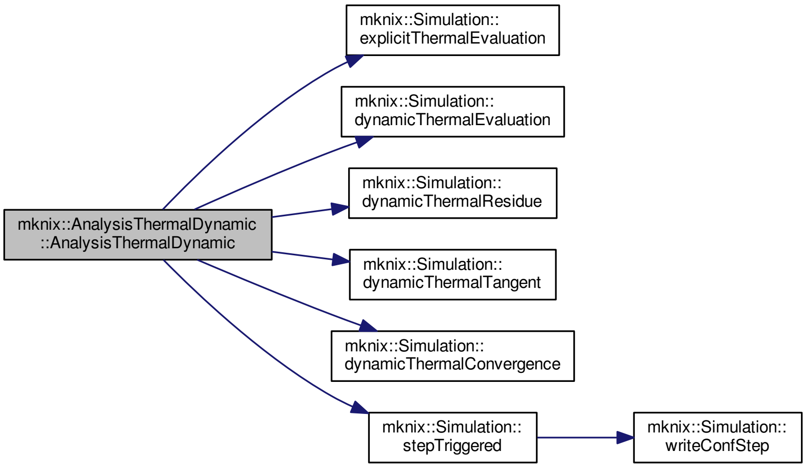
6.4.2.1 mknix::AnalysisThermalDynamic::AnalysisThermalDynamic ( )

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:


mknix::AnalysisThermalDynamic ::AnalysisThermalDynamic
mknix::Simulation:: explicitThermalEvaluation
mknix::Simulation:: dynamicThermalEvaluation
mknix::Simulation:: dynamicThermalResidue
mknix::Simulation:: dynamicThermalTangent
mknix::Simulation:: dynamicThermalConvergence
mknix::Simulation:: stepTriggered
mknix::Simulation:: writeConfStep

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 (lmx::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 (lmx::Vector< data_type > ∗ *qt_in*, lmx::Vector< data_type > ∗ *qdot_in* = 0, lmx::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.

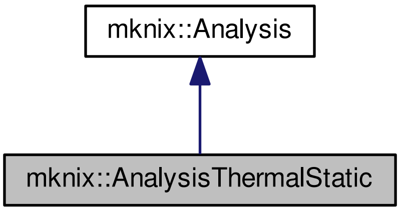
The documentation for this class was generated from the following files:

- analysisthermaldynamic.h
- analysisthermaldynamic.cpp

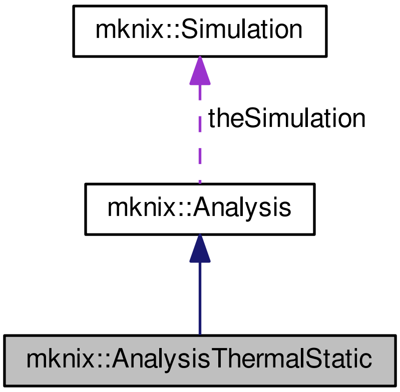
6.5 mknix::AnalysisThermalStatic Class Reference

```
#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

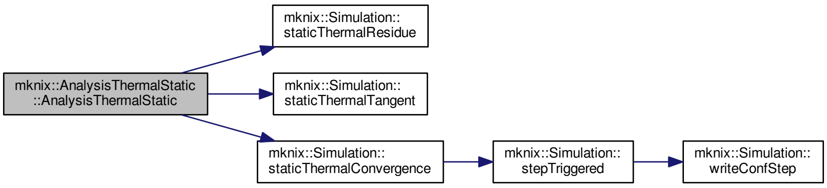
6.5.2.1 mknix::AnalysisThermalStatic::AnalysisThermalStatic ( )

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 (**lmx::Vector**< **data_type** > ∗ *q_in,* **lmx::Vector**< **data_type** > ∗ *qdot_in* = 0, **lmx::Vector**< **data_type** > ∗ *not_used* = 0) `[virtual]`

Implements mknix::Analysis.

Definition at line 48 of file analysisthermalstatic.cpp.

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

Implements mknix::Analysis.

Definition at line 39 of file analysisthermalstatic.h.

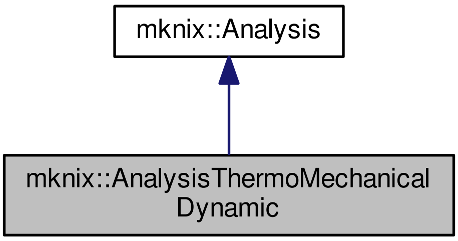
The documentation for this class was generated from the following files:

- analysisthermalstatic.h
- analysisthermalstatic.cpp

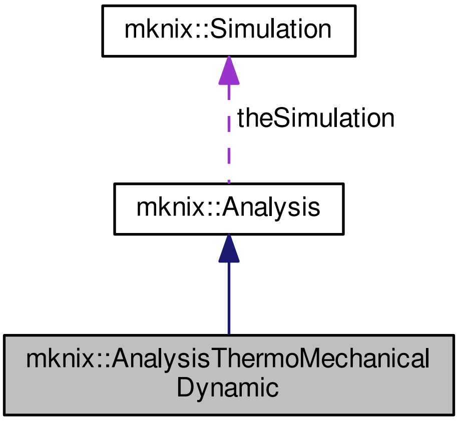
6.6 mknix::AnalysisThermoMechanicalDynamic Class Reference

```
#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 analysisthermomechanicaldynamic.h.

6.6.2 Constructor & Destructor Documentation

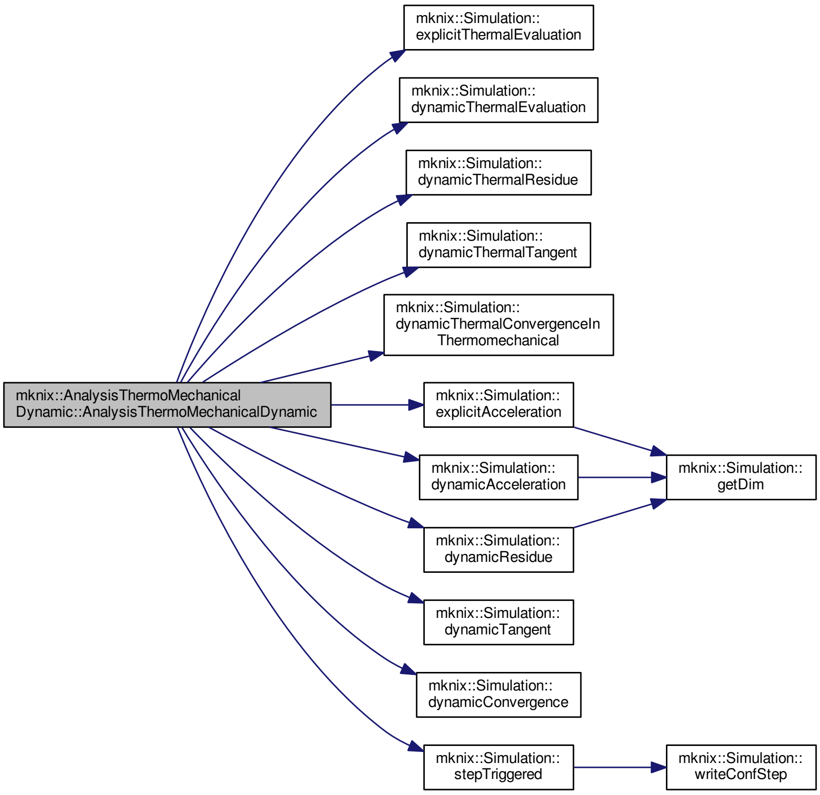
6.6.2.1 mknix::AnalysisThermoMechanicalDynamic::AnalysisThermoMechanicalDynamic ( )

Definition at line 25 of file analysisthermomechanicaldynamic.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 analysisthermomechanicaldynamic.cpp.

Here is the call graph for this function:


mknix::Simulation:: explicitThermalEvaluation
mknix::Simulation:: dynamicThermalEvaluation
mknix::Simulation:: dynamicThermalResidue
mknix::Simulation:: dynamicThermalTangent
mknix::Simulation:: dynamicThermalConvergenceIn Thermomechanical
mknix::AnalysisThermoMechanical Dynamic::AnalysisThermoMechanicalDynamic
mknix::Simulation:: explicitAcceleration
mknix::Simulation:: dynamicAcceleration
mknix::Simulation:: getDim
mknix::Simulation:: dynamicResidue
mknix::Simulation:: dynamicTangent
mknix::Simulation:: dynamicConvergence
mknix::Simulation:: stepTriggered
mknix::Simulation:: writeConfStep

6.6.2.3 mknix::AnalysisThermoMechanicalDynamic::∼AnalysisThermoMechanicalDynamic ( )

Definition at line 84 of file analysisthermomechanicaldynamic.cpp.

6.6.3 Member Function Documentation

6.6.3.1 void mknix::AnalysisThermoMechanicalDynamic::solve (lmx::Vector< data_type > * *qt_in*, lmx::Vector< data_type > * *q_in*, lmx::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.

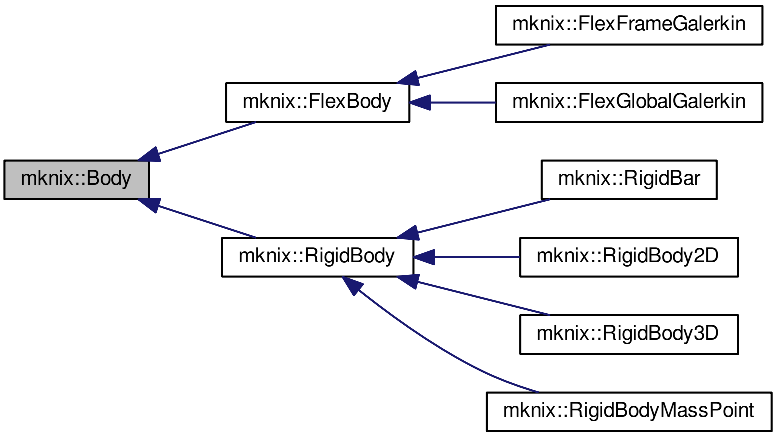
The documentation for this class was generated from the following files:

- analysisthermomechanicaldynamic.h
- analysisthermomechanicaldynamic.cpp

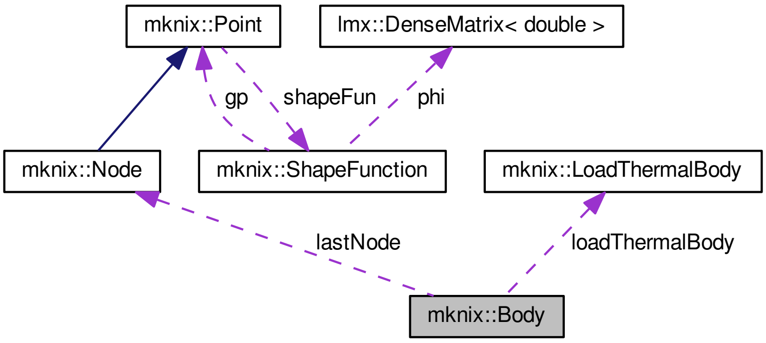
6.7 mknix::Body Class Reference

```
#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 (lmx::Matrix< data_type > &)

  Assembles the local conductivity into the global matrix by calling each cell's cascade function.
- virtual void assembleConductivityMatrix (lmx::Matrix< data_type > &)

  Assembles the local conductivity into the global matrix by calling each cell's cascade function.
- virtual void assembleExternalHeat (lmx::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 (lmx::Matrix< data_type > &)=0
- virtual void assembleExternalForces (lmx::Vector< data_type > &)=0
- void setTemperature (double)
- virtual void setMechanical ()
- virtual void setOutput (std::string)=0
- virtual void outputStep (const lmx::Vector< data_type > &, const lmx::Vector< data_type > &)=0
- virtual void outputStep (const lmx::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< lmx::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

title_in	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]

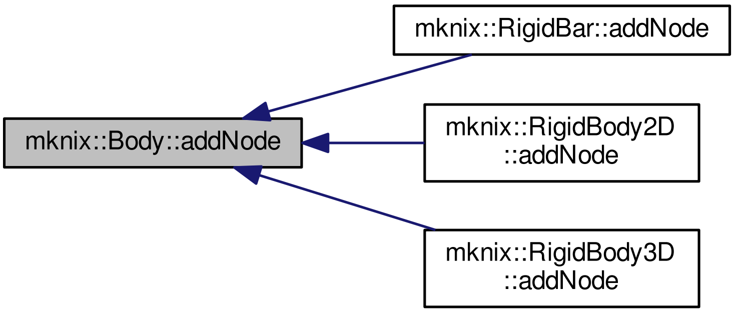
Definition at line 173 of file body.h.

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 (lmx::Matrix< data_type > & *globalCapacity*) [virtual]

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

Parameters

globalCapacity	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 (lmx::Matrix< data_type > & *globalConductivity*) [virtual]

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

Parameters

global↩ Conductivity	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 (lmx::Vector< data_type > &) [pure virtual]

Implemented in mknix::FlexFrameGalerkin, mknix::FlexGlobalGalerkin, and mknix::RigidBody.

6.7.3.12 void mknix::Body::assembleExternalHeat (lmx::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 (lmx::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.

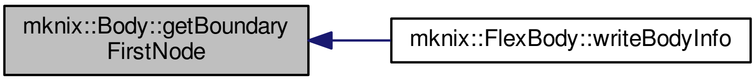
6.7.3.18 virtual void mknix::Body::calcMassMatrix ( ) [pure virtual]

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:


mknix::Body::getBoundary FirstNode
mknix::FlexBody::writeBodyInfo

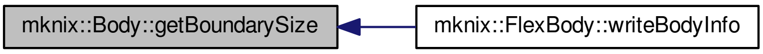
6.7.3.20 virtual Point* mknix::Body::getBoundaryNextNode (Point * *node_in*) [inline], [virtual]

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:



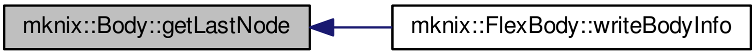
6.7.3.22 int mknix::Body::getCellLastNumber ( ) [inline]

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.

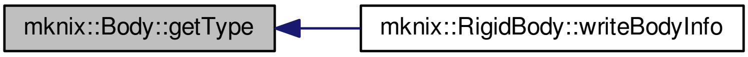
6.7.3.26 virtual int mknix::Body::getNodesSize ( ) [inline], [virtual]

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::RigidBar, 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.

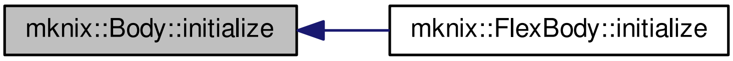
Returns

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 lmx::Vector< data_type > & , const lmx::Vector< data_type > &) [pure virtual]

Implemented in mknix::FlexFrameGalerkin, mknix::FlexGlobalGalerkin, and mknix::RigidBody.

6.7.3.30 virtual void mknix::Body::outputStep (const lmx::Vector< data_type > &) [pure virtual]

Implemented in mknix::FlexFrameGalerkin, mknix::FlexGlobalGalerkin, and mknix::RigidBody.

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

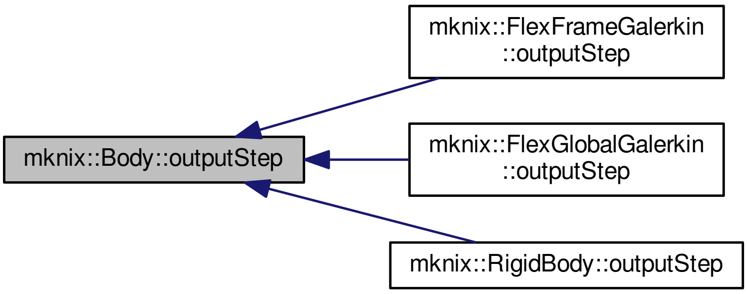
Postprocess and store thermal step results for any analysis.

Returns

void

Definition at line 232 of file body.cpp.

Here is the caller graph for this function:


mknix::FlexFrameGalerkin ::outputStep
mknix::Body::outputStep
mknix::FlexGlobalGalerkin ::outputStep
mknix::RigidBody::outputStep

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

Streams the data stored during the analysis to a file.

Parameters

outFile	Output files

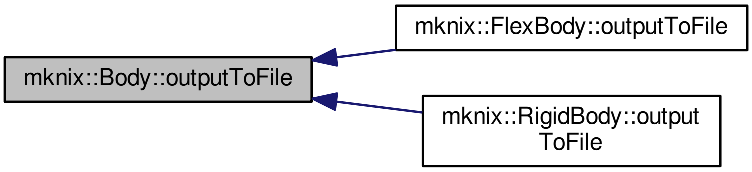
Returns

void

Reimplemented in mknix::FlexBody, and mknix::RigidBody.

Definition at line 267 of file body.cpp.

Here is the caller graph for this function:


mknix::Body::outputToFile
mknix::FlexBody::outputToFile
mknix::RigidBody::output ToFile

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<lmx::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.

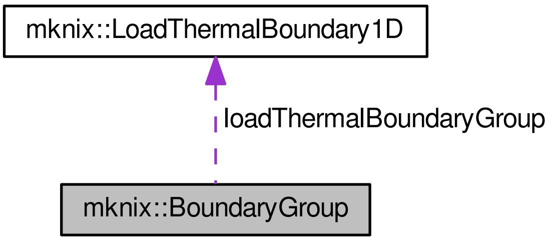
The documentation for this class was generated from the following files:

- 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 (lmx::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

title_in	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 (lmx::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.

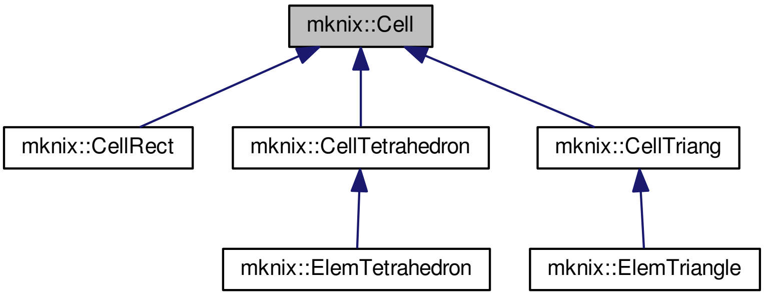
The documentation for this class was generated from the following files:

- boundarygroup.h
- boundarygroup.cpp

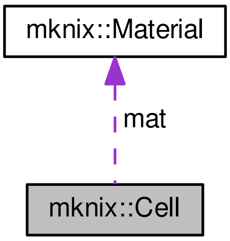
6.9 mknix::Cell Class Reference

`#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 (lmx::Matrix< data_type > &)
- void computeConductivityGaussPoints ()
- void assembleConductivityGaussPoints (lmx::Matrix< data_type > &)
- void computeQextGaussPoints (LoadThermalBody ∗)
- void assembleQextGaussPoints (lmx::Vector< data_type > &)
- void computeMGaussPoints ()
- void assembleMGaussPoints (lmx::Matrix< data_type > &)
- void computeFintGaussPoints ()
- void computeNLFintGaussPoints ()
- void assembleFintGaussPoints (lmx::Vector< data_type > &)
- void computeFextGaussPoints ()
- void assembleFextGaussPoints (lmx::Vector< data_type > &)
- void computeKGaussPoints ()
- void computeNLKGaussPoints ()
- void assembleKGaussPoints (lmx::Matrix< data_type > &)
- void assembleRGaussPoints (lmx::Vector< data_type > &, int)
- void assembleNLRGaussPoints (lmx::Vector< data_type > &, int)
- double calcPotentialEGaussPoints (const lmx::Vector< data_type > &)
- double calcKineticEGaussPoints (const lmx::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 (lmx::Matrix< data_type > & *globalCapacity*)

Definition at line 87 of file cell.cpp.

6.9.3.2 void mknix::Cell::assembleConductivityGaussPoints (lmx::Matrix< data_type > & *globalConductivity*)

Definition at line 102 of file cell.cpp.

6.9.3.3 void mknix::Cell::assembleFextGaussPoints (lmx::Vector< data_type > & *globalFext*)

Definition at line 173 of file cell.cpp.

6.9.3.4 void mknix::Cell::assembleFintGaussPoints (lmx::Vector< data_type > & *globalFint*)

Definition at line 157 of file cell.cpp.

6.9.3.5 void mknix::Cell::assembleKGaussPoints (lmx::Matrix< data_type > & *globalTangent*)

Definition at line 197 of file cell.cpp.

6.9.3.6 void mknix::Cell::assembleMGaussPoints (lmx::Matrix< data_type > & *globalMass*)

Definition at line 133 of file cell.cpp.

6.9.3.7 void mknix::Cell::assembleNLRGaussPoints (lmx::Vector< data_type > & *globalStress,* int *firstNode*)

Definition at line 216 of file cell.cpp.

6.9.3.8 void mknix::Cell::assembleQextGaussPoints (lmx::Vector< data_type > & *globalQext*)

Definition at line 117 of file cell.cpp.

6.9.3.9 void mknix::Cell::assembleRGaussPoints (lmx::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 lmx::Vector< data_type > & *qdot*)

Definition at line 238 of file cell.cpp.

6.9.3.12 double mknix::Cell::calcPotentialEGaussPoints (const lmx::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.

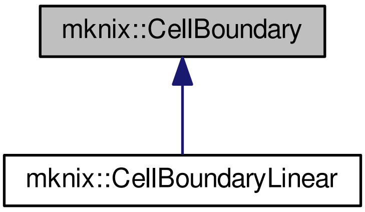
The documentation for this class was generated from the following files:

- 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 (lmx::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 (lmx::Vector< data_type > & *globalQext*)

Definition at line 65 of file cellboundary.cpp.

6.10.3.2 void mknix::CellBoundary::computeQextGaussPoints (LoadThermalBoundary1D ∗ *loadThermalBoundary1D_in*)

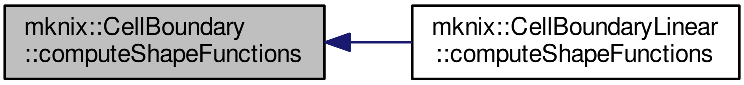
Definition at line 58 of file cellboundary.cpp.

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:

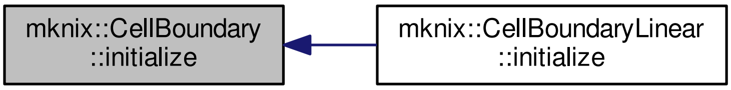

mknix::CellBoundary ::computeShapeFunctions
mknix::CellBoundaryLinear ::computeShapeFunctions

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:


mknix::CellBoundary ::initialize
mknix::CellBoundaryLinear ::initialize

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.

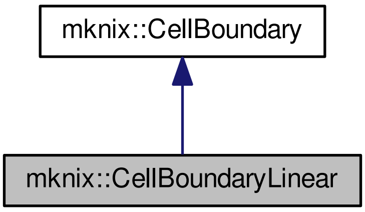
The documentation for this class was generated from the following files:

- cellboundary.h
- cellboundary.cpp

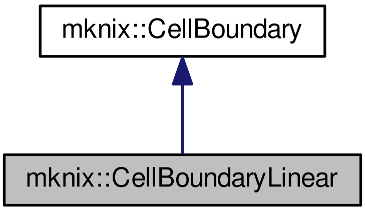
6.11 mknix::CellBoundaryLinear Class Reference

```
#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

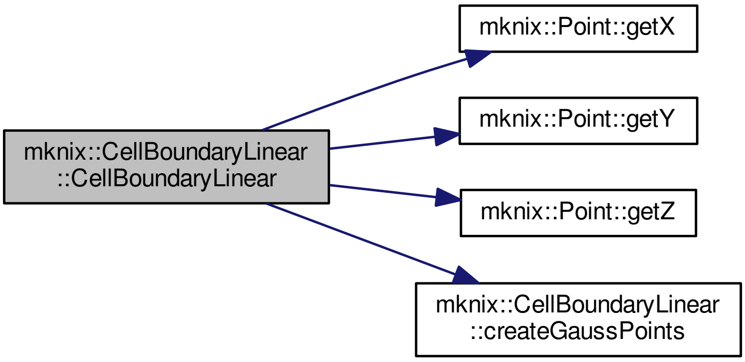
6.11.2.1 mknix::CellBoundaryLinear::CellBoundaryLinear ( )

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.

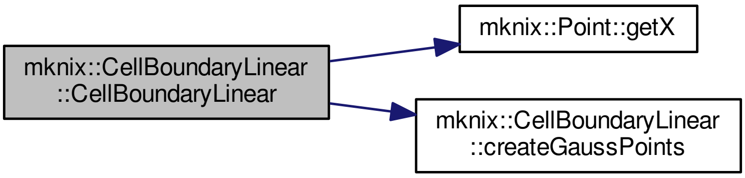
Here is the call graph for this function:



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.

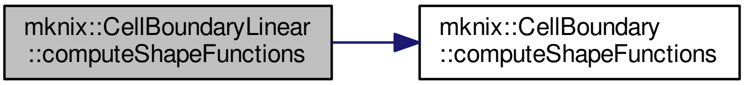
6.11.3 Member Function Documentation

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

Reimplemented from mknix::CellBoundary.

Definition at line 92 of file cellboundarylinear.cpp.

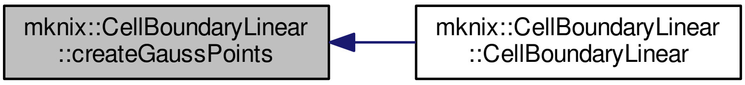
Here is the call graph for this function:


mknix::CellBoundaryLinear ::computeShapeFunctions
mknix::CellBoundary ::computeShapeFunctions

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:

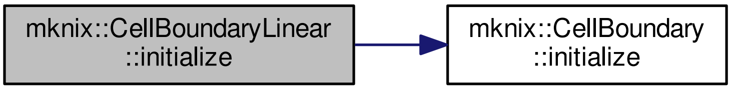

mknix::CellBoundaryLinear ::createGaussPoints
mknix::CellBoundaryLinear ::CellBoundaryLinear

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:


mknix::CellBoundaryLinear ::initialize
mknix::CellBoundary ::initialize

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.

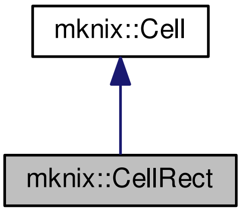
The documentation for this class was generated from the following files:

- cellboundarylinear.h
- cellboundarylinear.cpp

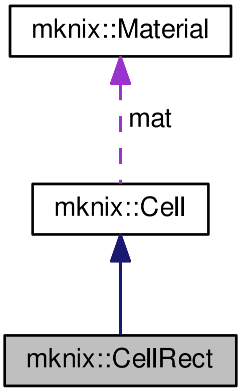
6.12 mknix::CellRect Class Reference

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

Inheritance diagram for mknix::CellRect:



Collaboration diagram for mknix::CellRect:



Public Member Functions

- CellRect ()
- CellRect (Material &, std::string, double, int, double, double, double, double, double, double, double, double, double, double, double, double, double, double)
- CellRect (Material &, std::string, double, int, double)
- ∼CellRect ()
- void initialize (std::vector< Node ∗ > &)
- void gnuplotOut (std::ofstream &, std::ofstream &)

Additional Inherited Members

6.12.1 Detailed Description

Author

Daniel Iglesias

Definition at line 47 of file cellrect.h.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 mknix::CellRect::CellRect ( )

Definition at line 9 of file cellrect.cpp.

6.12.2.2 mknix::CellRect::CellRect (Material & *material_in*, std::string *formulation_in*, double *alpha_in*, int *nGPoints_in*, double *x1_in*, double *y1_in*, double *x2_in*, double *y2_in*, double *x3_in*, double *y3_in*, double *x4_in*, double *y4_in*, double *dcx_in*, double *dcy_in*, double *minX_in*, double *minY_in*, double *maxX_in*, double *maxY_in*)

Definition at line 14 of file cellrect.cpp.

6.12.2.3 mknix::CellRect::CellRect (Material & *material_in*, std::string *formulation_in*, double *alpha_in*, int *nGPoints_in*, double *x1_in*, double *y1_in*, double *z1_in*, double *x2_in*, double *y2_in*, double *z2_in*, double *x3_in*, double *y3_in*, double *z3_in*, double *x4_in*, double *y4_in*, double *z4_in*, double *x5_in*, double *y5_in*, double *z5_in*, double *x6_in*, double *y6_in*, double *z6_in*, double *x7_in*, double *y7_in*, double *z7_in*, double *x8_in*, double *y8_in*, double *z8_in*, double *dcx_in*, double *dcy_in*, double *dcz_in*, double *minX_in*, double *minY_in*, double *minZ_in*, double *maxX_in*, double *maxY_in*, double *maxZ_in*)

Definition at line 55 of file cellrect.cpp.

6.12.2.4 mknix::CellRect::∼CellRect ( )

Definition at line 116 of file cellrect.cpp.

6.12.3 Member Function Documentation

6.12.3.1 void mknix::CellRect::gnuplotOut (std::ofstream & *data*, std::ofstream & *gpdata*) [virtual]

Implements mknix::Cell.

Definition at line 309 of file cellrect.cpp.

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.

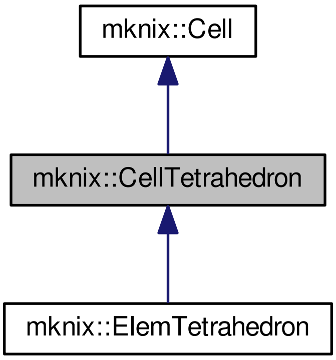
The documentation for this class was generated from the following files:

- cellrect.h
- cellrect.cpp

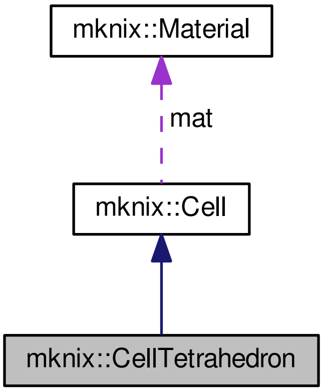
6.13 mknix::CellTetrahedron Class Reference

`#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

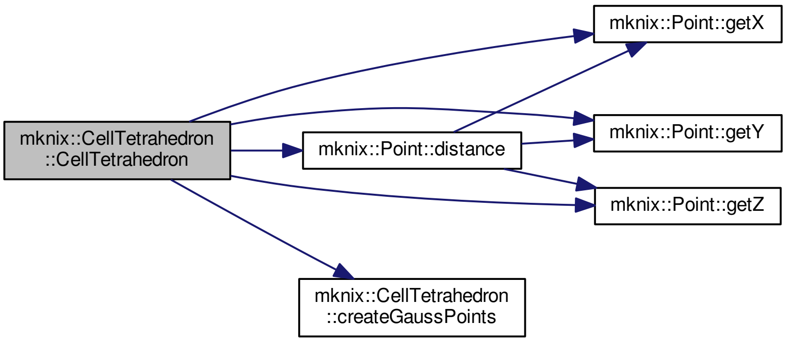
6.13.2.1 mknix::CellTetrahedron::CellTetrahedron ( )

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:


mknix::Point::getX
mknix::Point::getY
mknix::CellTetrahedron ::CellTetrahedron
mknix::Point::distance
mknix::Point::getZ
mknix::CellTetrahedron ::createGaussPoints

6.13.2.3 mknix::CellTetrahedron::∼CellTetrahedron ( ) [virtual]

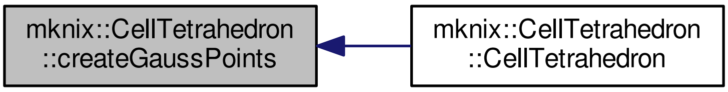
Definition at line 65 of file celltetrahedron.cpp.

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 & *data*, std::ofstream & *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<3, double> mknix::CellTetrahedron::points [protected]

position of vertex points

Definition at line 46 of file celltetrahedron.h.

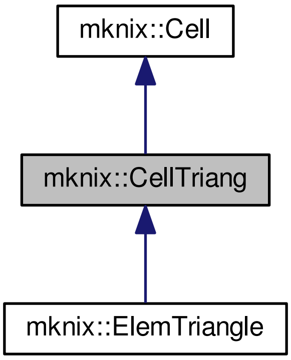
The documentation for this class was generated from the following files:

- celltetrahedron.h
- celltetrahedron.cpp

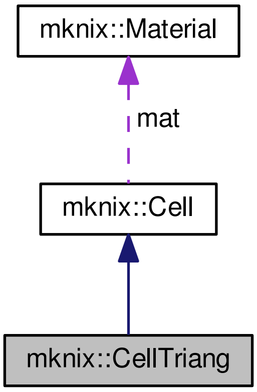
6.14 mknix::CellTriang Class Reference

```
#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 & Destructor Documentation

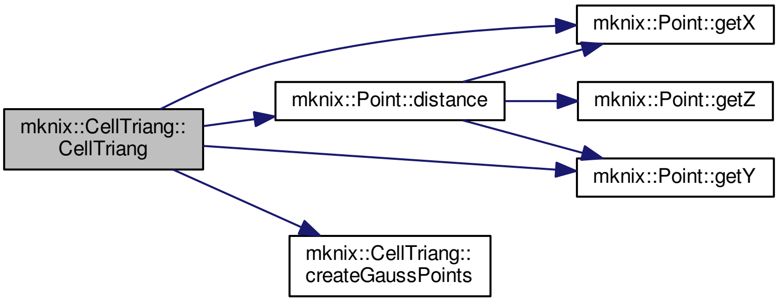
6.14.2.1 mknix::CellTriang::CellTriang ( )

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.

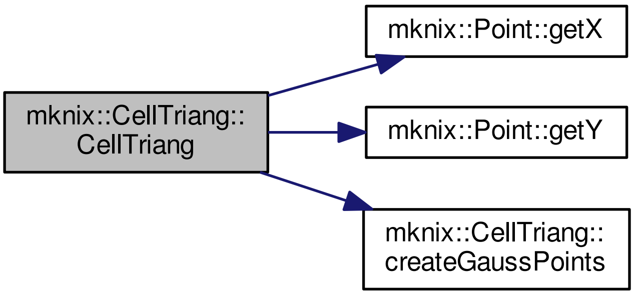
Here is the call graph for this function:



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]

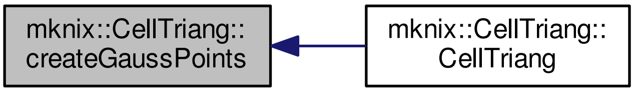
Definition at line 81 of file celltriang.cpp.

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 ( )

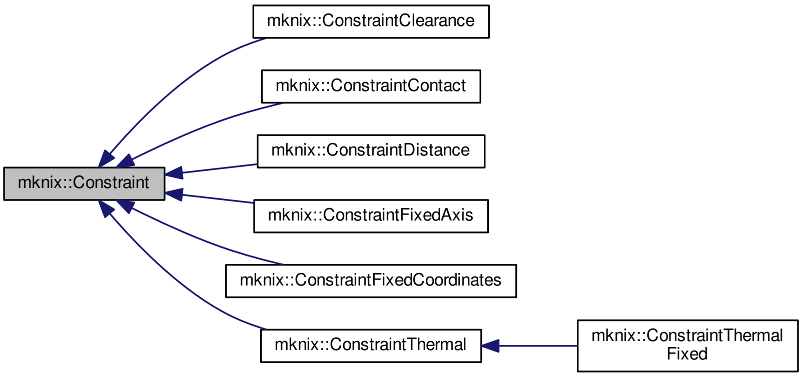
The documentation for this class was generated from the following file:

- compbar.h

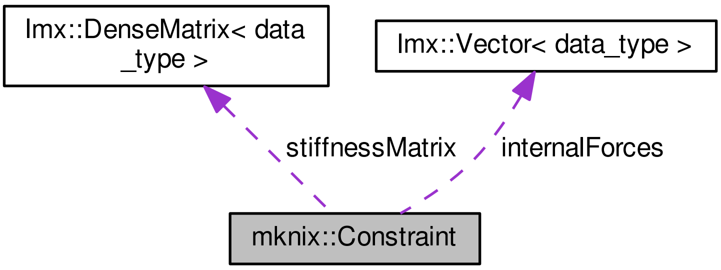
6.16 mknix::Constraint Class Reference

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

Inheritance diagram for mknix::Constraint:


mknix::Constraint
mknix::ConstraintClearance
mknix::ConstraintContact
mknix::ConstraintDistance
mknix::ConstraintFixedAxis
mknix::ConstraintFixedCoordinates
mknix::ConstraintThermal
mknix::ConstraintThermal Fixed

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 (lmx::Vector< data_type > &)
- virtual void assembleTangentMatrix (lmx::Matrix< data_type > &)
- virtual bool checkAugmented ()
- virtual void clearAugmented ()

- virtual Node ∗ getNode (int nodeNumber)
- void outputStep (const lmx::Vector< data_type > &, const lmx::Vector< data_type > &)
- void outputStep (const lmx::Vector< data_type > &)
- void outputToFile (std::ofstream ∗)

Protected Attributes

- int dim
- double alpha
- std::string method
- std::string title
- std::vector< Node ∗ > nodes
- lmx::Vector< data_type > internalForces
- std::vector< lmx::Vector< data_type > > internalForcesOutput
- lmx::DenseMatrix< data_type > stiffnessMatrix
- std::vector< double > lambda
- std::vector< double > phi
- std::vector< lmx::Vector< data_type > > phi_q
- std::vector< lmx::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.

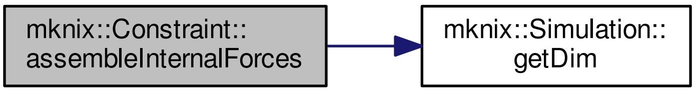
6.16.3 Member Function Documentation

6.16.3.1 void mknix::Constraint::assembleInternalForces (lmx::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:

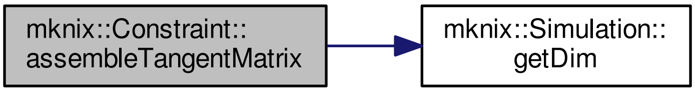

mknix::Constraint:: assembleInternalForces
mknix::Simulation:: getDim

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

Reimplemented in mknix::ConstraintThermal.

Definition at line 129 of file constraint.cpp.

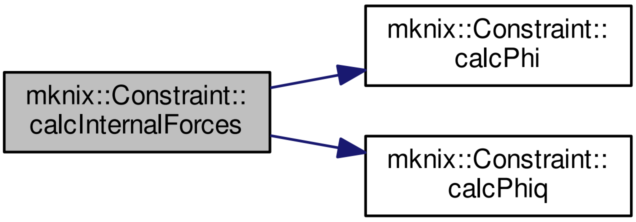
Here is the call graph for this function:


mknix::Constraint:: assembleTangentMatrix
mknix::Simulation:: getDim

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

Definition at line 67 of file constraint.cpp.

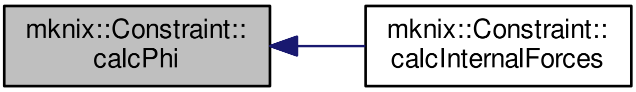
Here is the call graph for this function:


mknix::Constraint:: calcInternalForces
mknix::Constraint:: calcPhi
mknix::Constraint:: calcPhiq

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.

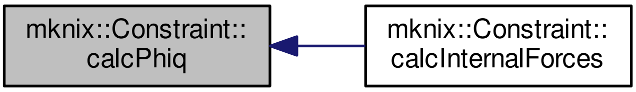
Here is the caller graph for this function:



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.

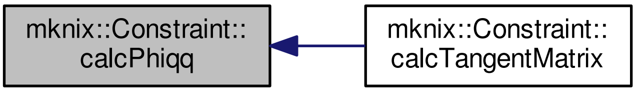
Here is the caller graph for this function:



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.

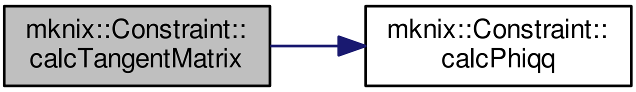
Here is the caller graph for this function:



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 lmx::Vector< data_type > & *q,* const lmx::Vector< data_type > & *qdot*)

Definition at line 218 of file constraint.cpp.

6.16.3.12 void mknix::Constraint::outputStep (const lmx::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 lmx::Vector<data_type> mknix::Constraint::internalForces [protected]

Definition at line 82 of file constraint.h.

6.16.4.4 std::vector< lmx::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< lmx::Vector<data_type> > mknix::Constraint::phi_q [protected]

Definition at line 87 of file constraint.h.

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

Definition at line 88 of file constraint.h.

6.16.4.11 lmx::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.

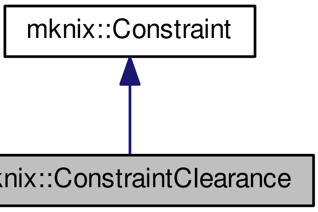
The documentation for this class was generated from the following files:

- constraint.h
- constraint.cpp

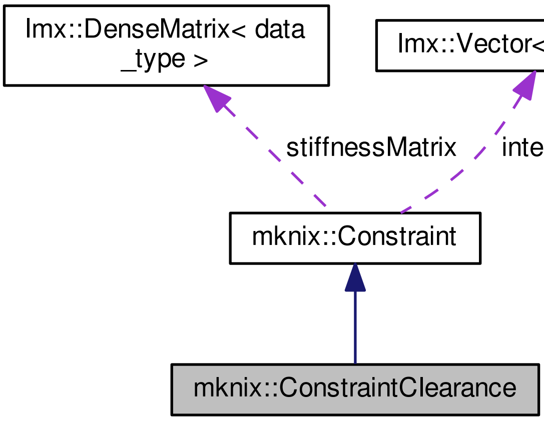
6.17 mknix::ConstraintClearance Class Reference

```
#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 ()
- lmx::Vector< data_type > & getInternalForces ()
- lmx::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 lmx::Vector<data_type>& mknix::ConstraintClearance::getInternalForces ( ) [inline]

Definition at line 45 of file constraintclearance.h.

6.17.3.5 lmx::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.

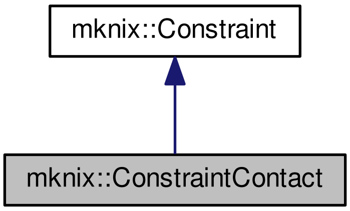
The documentation for this class was generated from the following files:

- constraintclearance.h
- constraintclearance.cpp

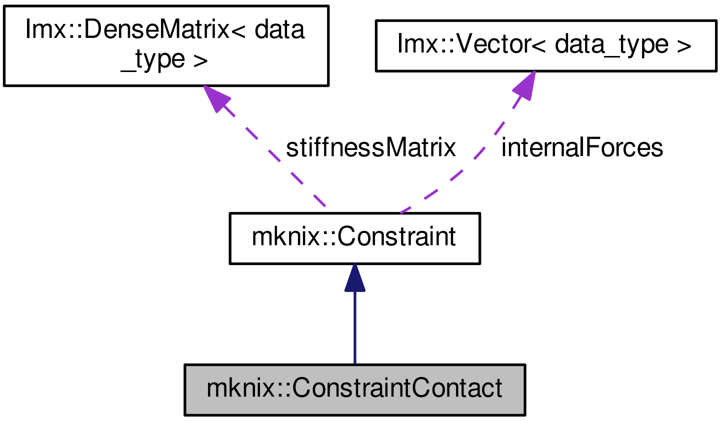
6.18 mknix::ConstraintContact Class Reference

```
#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 ()
- lmx::Vector< data_type > & getInternalForces ()
- lmx::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 lmx::Vector<data_type>& mknix::ConstraintContact::getInternalForces ( ) [inline]

Definition at line 45 of file constraintcontact.h.

6.18.3.6 lmx::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.

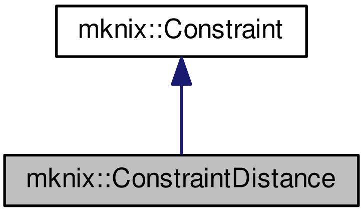
The documentation for this class was generated from the following files:

- constraintcontact.h
- constraintcontact.cpp

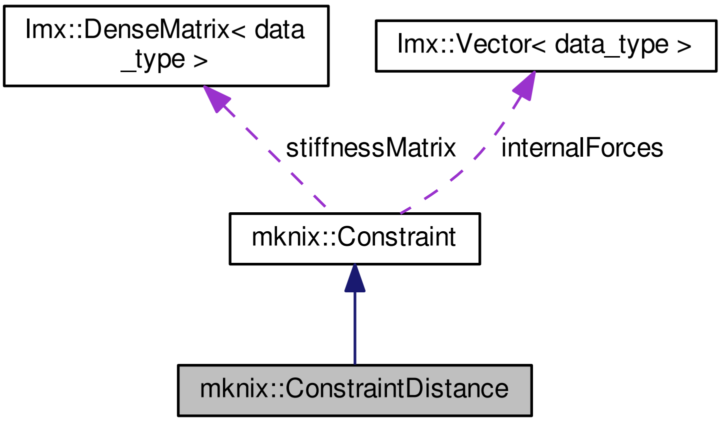
6.19 mknix::ConstraintDistance Class Reference

```
#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)
- lmx::Vector< data_type > & getInternalForces ()
- lmx::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.

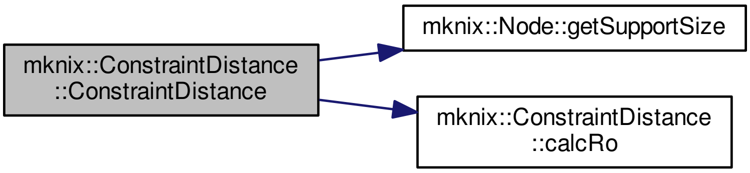
6.19.2.2 mknix::ConstraintDistance::∼ConstraintDistance ( )

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:


mknix::ConstraintDistance ::ConstraintDistance
mknix::Node::getSupportSize
mknix::ConstraintDistance ::calcRo

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]

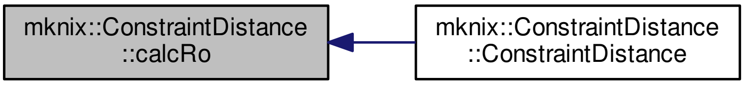
Implements mknix::Constraint.

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:


mknix::ConstraintDistance ::calcRo
mknix::ConstraintDistance ::ConstraintDistance

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

Definition at line 52 of file constraintdistance.h.

6.19.3.6 lmx::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.

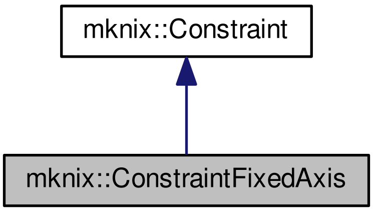
The documentation for this class was generated from the following files:

- constraintdistance.h
- constraintdistance.cpp

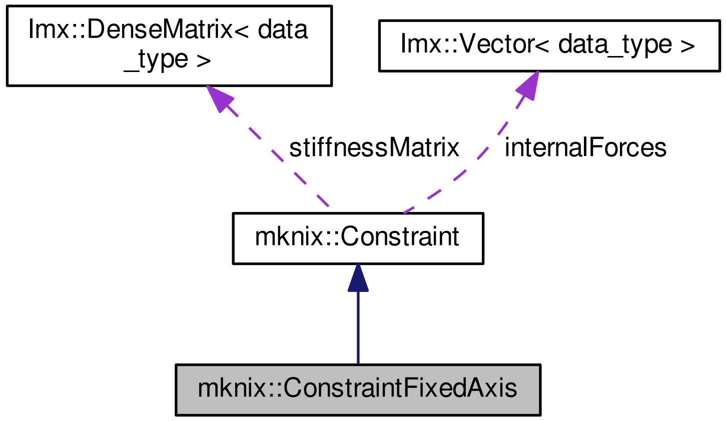
6.20 mknix::ConstraintFixedAxis Class Reference

```
#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 ()
- lmx::Vector< data_type > & getInternalForces ()
- lmx::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

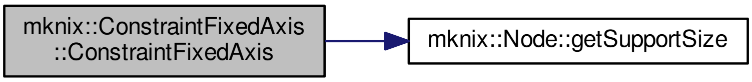
6.20.2.1 mknix::ConstraintFixedAxis::ConstraintFixedAxis ( )

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 lmx::Vector<data_type>& mknix::ConstraintFixedAxis::getInternalForces ( ) [inline]

Definition at line 45 of file constraintfixedaxis.h.

6.20.3.5 lmx::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.

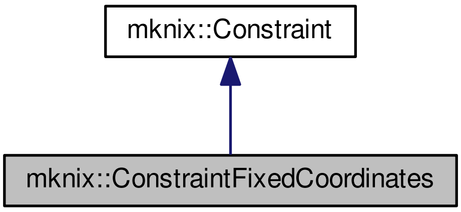
The documentation for this class was generated from the following files:

- constraintfixedaxis.h
- constraintfixedaxis.cpp

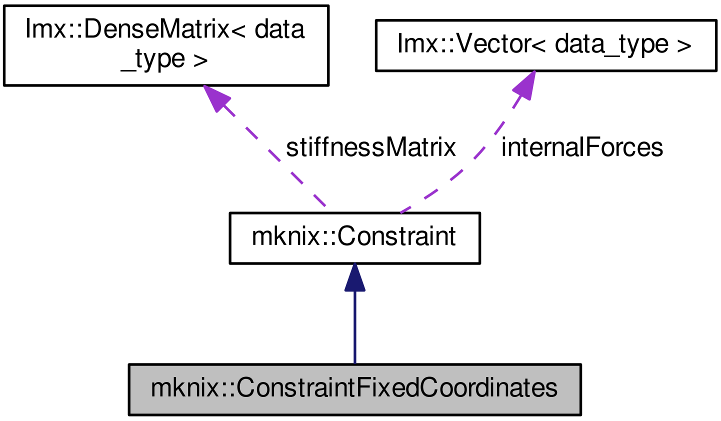
6.21 mknix::ConstraintFixedCoordinates Class Reference

`#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 ()
- lmx::Vector< data_type > & getInternalForces ()
- lmx::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

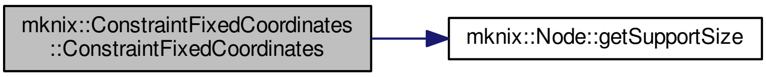
6.21.2.1 mknix::ConstraintFixedCoordinates::ConstraintFixedCoordinates ( )

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:


mknix::ConstraintFixedCoordinates
::ConstraintFixedCoordinates
mknix::Node::getSupportSize

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 lmx::Vector<data_type>& mknix::ConstraintFixedCoordinates::getInternalForces ( ) [inline]

Definition at line 45 of file constraintfixedcoordinates.h.

6.21.3.5 lmx::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.

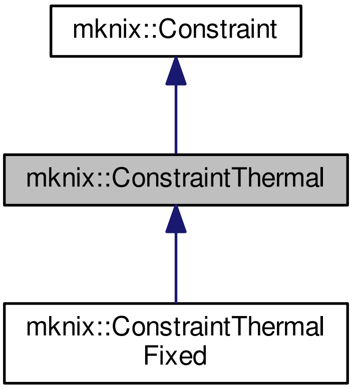
The documentation for this class was generated from the following files:

- constraintfixedcoordinates.h
- constraintfixedcoordinates.cpp

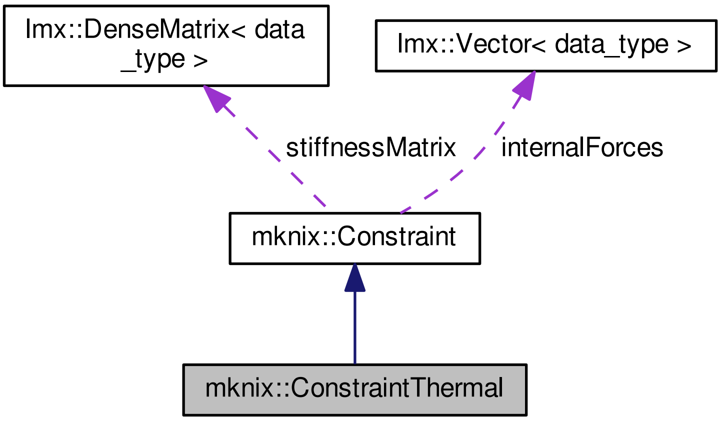
6.22 mknix::ConstraintThermal Class Reference

```
#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 (lmx::Vector< data_type > &)
- virtual void assembleTangentMatrix (lmx::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 (lmx::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 (lmx::Matrix< data_type > & *globalTangent*) `[virtual]`

Reimplemented from mknix::Constraint.

Definition at line 70 of file constraintthermal.cpp.

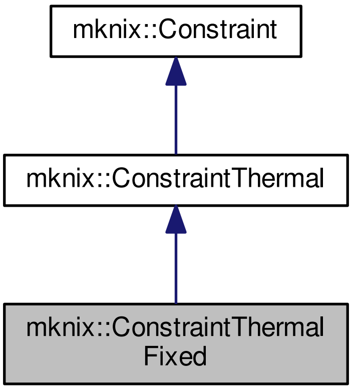
The documentation for this class was generated from the following files:

- constraintthermal.h
- constraintthermal.cpp

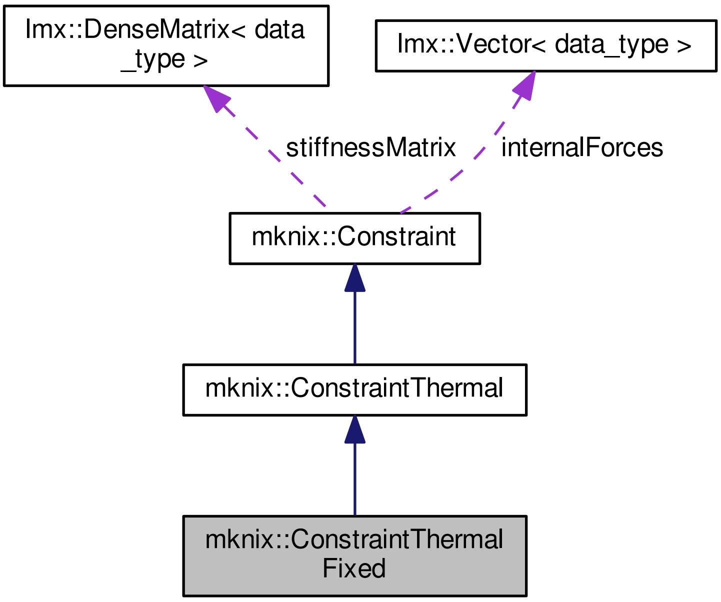
6.23 mknix::ConstraintThermalFixed Class Reference

`#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 ()
- lmx::Vector< data_type > & getInternalForces ()
- lmx::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

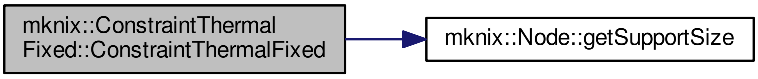
6.23.2.1 mknix::ConstraintThermalFixed::ConstraintThermalFixed ( )

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 lmx::Vector<data_type>& mknix::ConstraintThermalFixed::getInternalForces ( ) [inline]

Definition at line 45 of file constraintthermalfixed.h.

6.23.3.5 lmx::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 lmx::DenseMatrix< T > Class Template Reference

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

6.25.1 Detailed Description

template<typename T>class lmx::DenseMatrix< T >

Definition at line 39 of file cellrect.h.

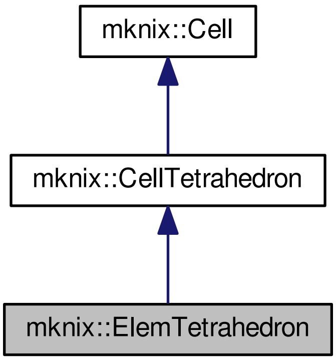
The documentation for this class was generated from the following file:

- cellrect.h

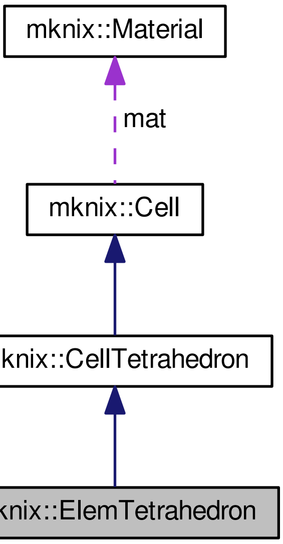
6.26 mknix::ElemTetrahedron Class Reference

```
#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]

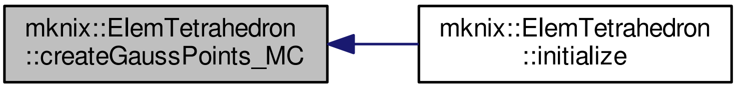
Reimplemented from mknix::Cell.

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:

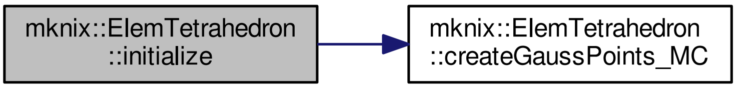


6.26.3.3 void mknix::ElemTetrahedron::initialize (std::vector< Node ∗ > & *nodes_in*) [virtual]

Reimplemented from mknix::Cell.

Definition at line 57 of file elemtetrahedron.cpp.

Here is the call graph for this function:


mknix::ElemTetrahedron ::initialize
mknix::ElemTetrahedron ::createGaussPoints_MC

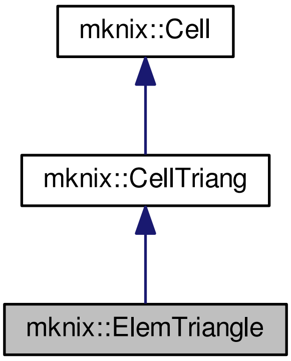
The documentation for this class was generated from the following files:

- elemtetrahedron.h
- elemtetrahedron.cpp

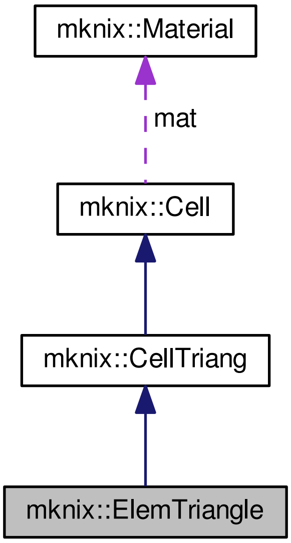
6.27 mknix::ElemTriangle Class Reference

```
#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]

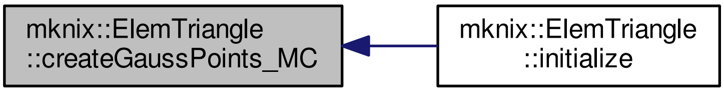
Reimplemented from mknix::Cell.

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:

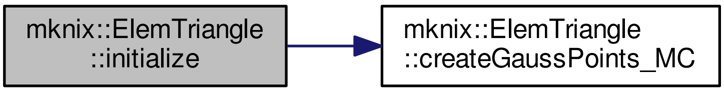

mknix::ElemTriangle ::createGaussPoints_MC
mknix::ElemTriangle ::initialize

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:


mknix::ElemTriangle ::initialize
mknix::ElemTriangle ::createGaussPoints_MC

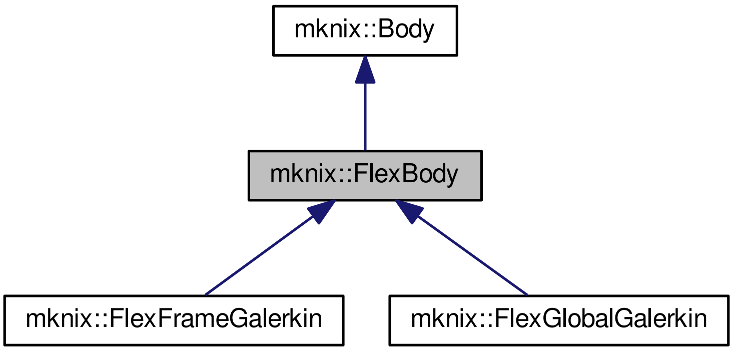
The documentation for this class was generated from the following files:

- elemtriangle.h
- elemtriangle.cpp

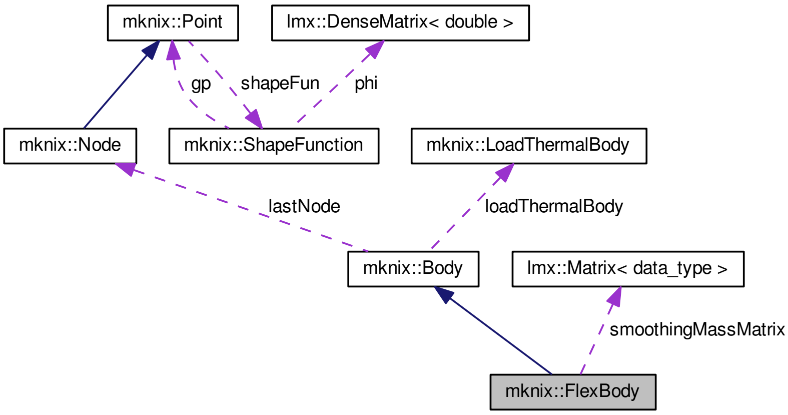
6.28 mknix::FlexBody Class Reference

`#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 (lmx::Vector< data_type > &)=0
- virtual void assembleTangentMatrix (lmx::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
- lmx::Matrix< data_type > smoothingMassMatrix
- std::vector< lmx::Vector< data_type > > stresses
- std::vector< lmx::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 (lmx::Vector< data_type > &) [pure virtual]

Implemented in mknix::FlexFrameGalerkin, and mknix::FlexGlobalGalerkin.

6.28.3.5 virtual void mknix::FlexBody::assembleTangentMatrix (lmx::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.

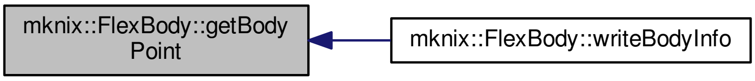
6.28.3.7 virtual void mknix::FlexBody::calcTangentMatrix ( ) [pure virtual]

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.

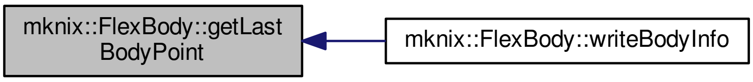
Here is the caller graph for this function:


mknix::FlexBody::getBody Point
mknix::FlexBody::writeBodyInfo

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:

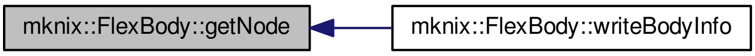

mknix::FlexBody::getLast BodyPoint
mknix::FlexBody::writeBodyInfo

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.

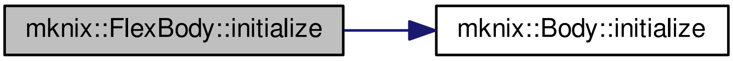
Returns

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

outFile	Output files

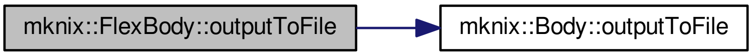
Returns

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

outputType_in	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]

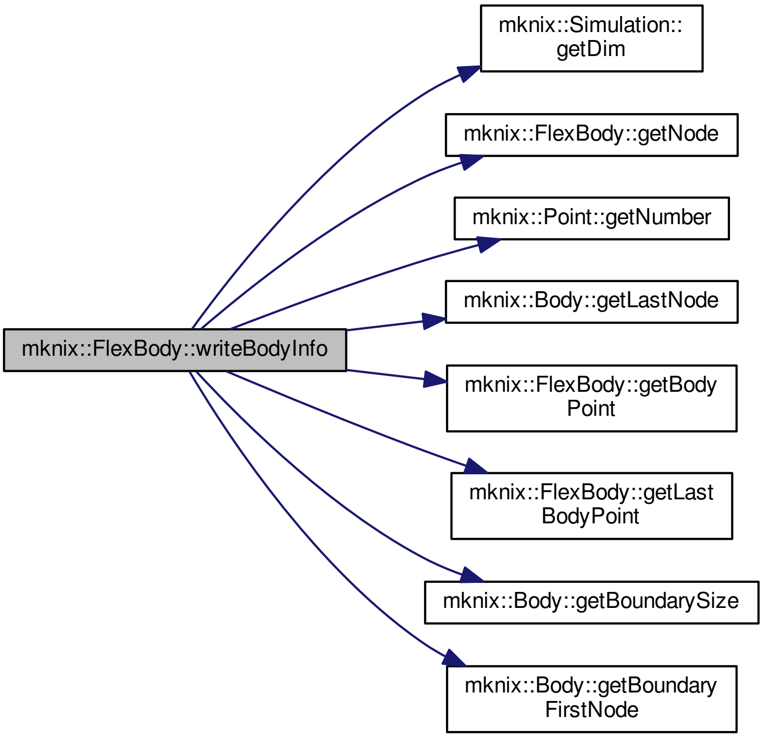
Implemented in mknix::FlexFrameGalerkin, and mknix::FlexGlobalGalerkin.

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:


mknix::Simulation::getDim
mknix::FlexBody::getNode
mknix::Point::getNumber
mknix::Body::getLastNode
mknix::FlexBody::writeBodyInfo
mknix::FlexBody::getBodyPoint
mknix::FlexBody::getLastBodyPoint
mknix::Body::getBoundarySize
mknix::Body::getBoundaryFirstNode

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< lmx::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 lmx::Matrix< data_type > mknix::FlexBody::smoothingMassMatrix [protected]

Definition at line 111 of file bodyflex.h.

6.28.4.8 std::vector< lmx::Vector<data_type> > mknix::FlexBody::stresses [protected]

Definition at line 112 of file bodyflex.h.

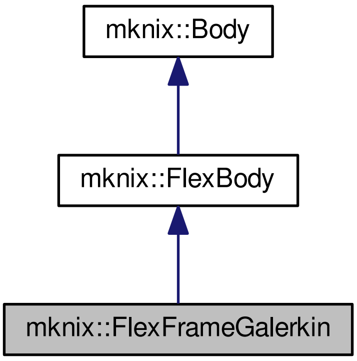
The documentation for this class was generated from the following files:

- bodyflex.h
- bodyflex.cpp

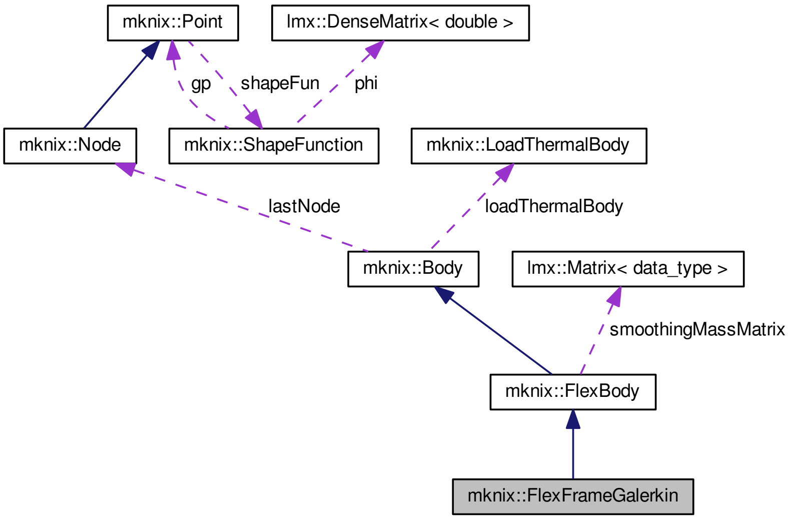
6.29 mknix::FlexFrameGalerkin Class Reference

```
#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 (lmx::Matrix< data_type > &)
- void assembleInternalForces (lmx::Vector< data_type > &)
- void assembleExternalForces (lmx::Vector< data_type > &)
- void assembleTangentMatrix (lmx::Matrix< data_type > &)
- void outputStep (const lmx::Vector< data_type > &, const lmx::Vector< data_type > &)

  Postprocess and store step results for dynamic analysis.
- void outputStep (const lmx::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 (lmx::Vector< data_type > & *globalExternalForces*) [virtual]

Implements mknix::Body.

Definition at line 151 of file bodyflexframegalerkin.cpp.

6.29.3.2 void mknix::FlexFrameGalerkin::assembleInternalForces (lmx::Vector< data_type > & *globalInternalForces*) [virtual]

Implements mknix::FlexBody.

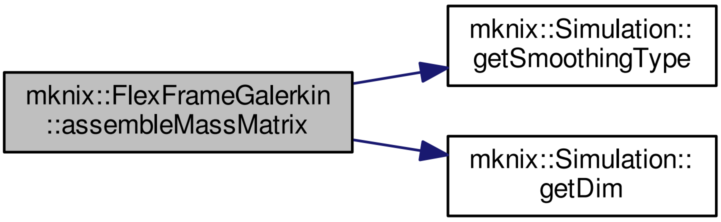
Definition at line 139 of file bodyflexframegalerkin.cpp.

6.29.3.3 void mknix::FlexFrameGalerkin::assembleMassMatrix (lmx::Matrix< data_type > & *globalMass*) [virtual]

Implements mknix::Body.

Definition at line 112 of file bodyflexframegalerkin.cpp.

Here is the call graph for this function:


mknix::FlexFrameGalerkin ::assembleMassMatrix
mknix::Simulation:: getSmoothingType
mknix::Simulation:: getDim

6.29.3.4 void mknix::FlexFrameGalerkin::assembleTangentMatrix (lmx::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 lmx::Vector< data_type > & *q*, const lmx::Vector< data_type > & *qdot*) [virtual]

Postprocess and store step results for dynamic analysis.

Parameters

q	Global configuration vector
qdot	Global configuration first derivative vector

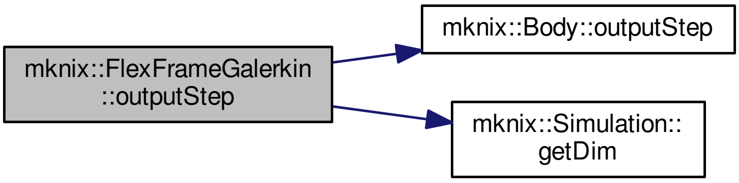
Returns

void

Implements mknix::Body.

Definition at line 180 of file bodyflexframegalerkin.cpp.

Here is the call graph for this function:


mknix::FlexFrameGalerkin ::outputStep
mknix::Body::outputStep
mknix::Simulation:: getDim

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

Postprocess and store step results for static analysis.

Parameters

q	Global configuration vector

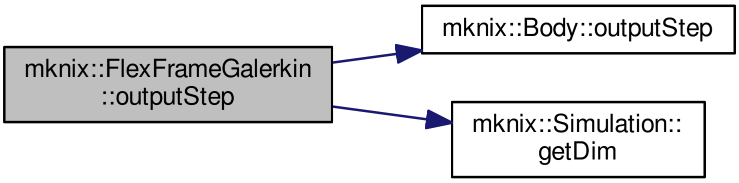
Returns

void

Implements mknix::Body.

Definition at line 237 of file bodyflexframegalerkin.cpp.

Here is the call graph for this function:


mknix::FlexFrameGalerkin ::outputStep
mknix::Body::outputStep
mknix::Simulation:: getDim

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.

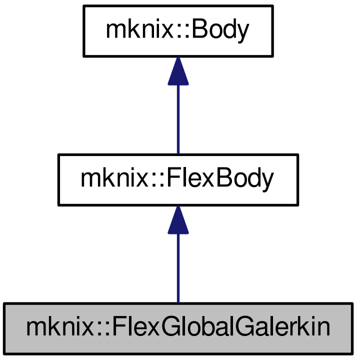
The documentation for this class was generated from the following files:

- bodyflexframegalerkin.h
- bodyflexframegalerkin.cpp

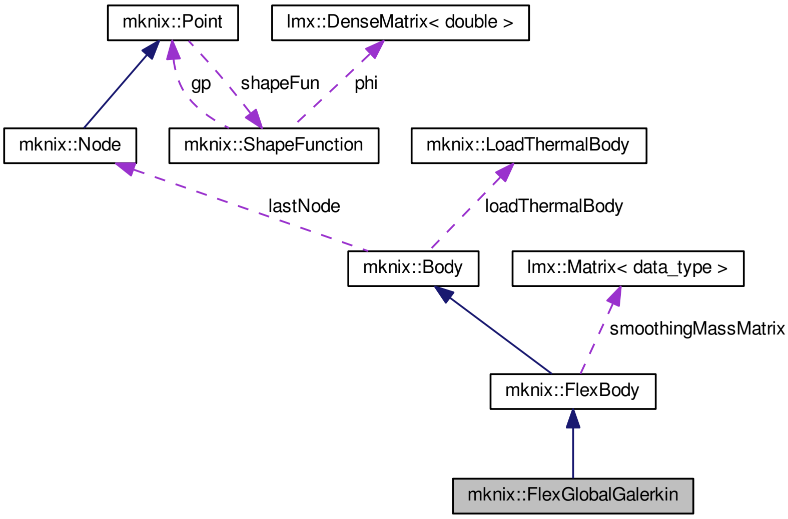
6.30 mknix::FlexGlobalGalerkin Class Reference

```
#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 (lmx::Matrix< data_type > &)
- void assembleInternalForces (lmx::Vector< data_type > &)
- void assembleExternalForces (lmx::Vector< data_type > &)
- void assembleTangentMatrix (lmx::Matrix< data_type > &)
- void outputStep (const lmx::Vector< data_type > &, const lmx::Vector< data_type > &)

  Postprocess and store step results for dynamic analysis.
- void outputStep (const lmx::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 (lmx::Vector< data_type > & *globalExternalForces*) [virtual]

Implements mknix::Body.

Definition at line 136 of file bodyflexglobalgalerkin.cpp.

6.30.3.2 void mknix::FlexGlobalGalerkin::assembleInternalForces (lmx::Vector< data_type > & *globalInternalForces*) [virtual]

Implements mknix::FlexBody.

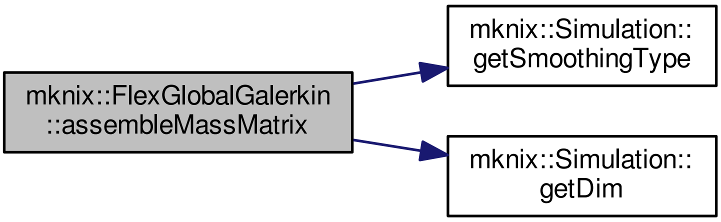
Definition at line 126 of file bodyflexglobalgalerkin.cpp.

6.30.3.3 void mknix::FlexGlobalGalerkin::assembleMassMatrix (lmx::Matrix< data_type > & *globalMass*) [virtual]

Implements mknix::Body.

Definition at line 101 of file bodyflexglobalgalerkin.cpp.

Here is the call graph for this function:


mknix::FlexGlobalGalerkin ::assembleMassMatrix
mknix::Simulation:: getSmoothingType
mknix::Simulation:: getDim

6.30.3.4 void mknix::FlexGlobalGalerkin::assembleTangentMatrix (lmx::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 lmx::Vector< data_type > & *q*, const lmx::Vector< data_type > & *qdot*) [virtual]

Postprocess and store step results for dynamic analysis.

Parameters

q	Global configuration vector
qdot	Global configuration first derivative vector

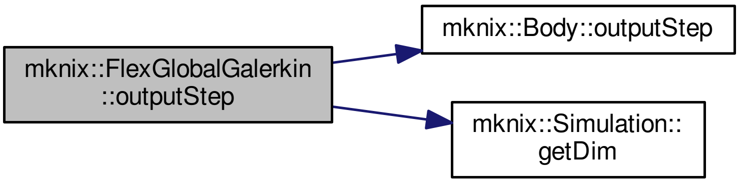
Returns

void

Implements mknix::Body.

Definition at line 161 of file bodyflexglobalgalerkin.cpp.

Here is the call graph for this function:


mknix::FlexGlobalGalerkin ::outputStep
mknix::Body::outputStep
mknix::Simulation:: getDim

6.30.3.11 void mknix::FlexGlobalGalerkin::outputStep (const lmx::Vector< data_type > & *q*) [virtual]

Postprocess and store step results for static analysis.

Parameters

q	Global configuration vector

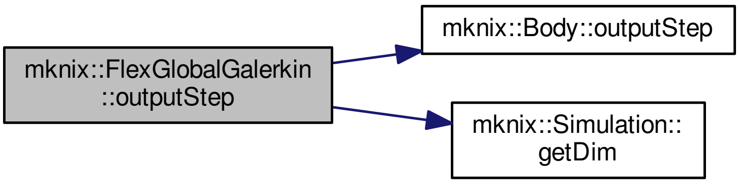
Returns

void

Implements mknix::Body.

Definition at line 212 of file bodyflexglobalgalerkin.cpp.

Here is the call graph for this function:


mknix::FlexGlobalGalerkin ::outputStep
mknix::Body::outputStep
mknix::Simulation:: getDim

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.

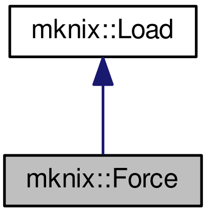
The documentation for this class was generated from the following files:

- bodyflexglobalgalerkin.h
- bodyflexglobalgalerkin.cpp

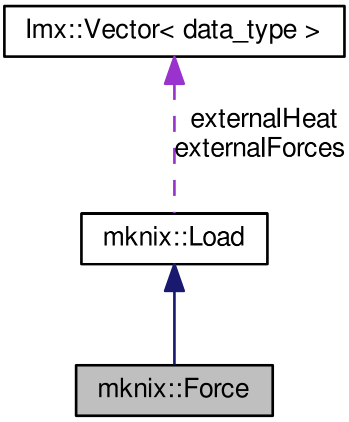
6.31 mknix::Force Class Reference

```
#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

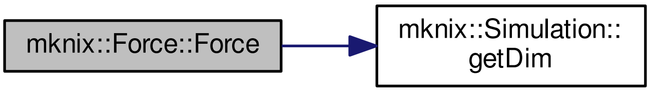
6.31.2.1 mknix::Force::Force ( )

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.

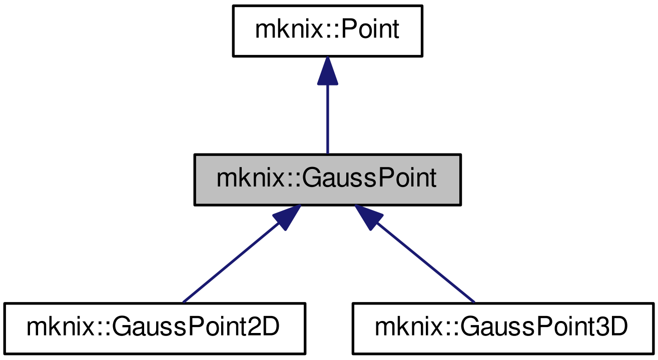
The documentation for this class was generated from the following files:

- force.h
- force.cpp

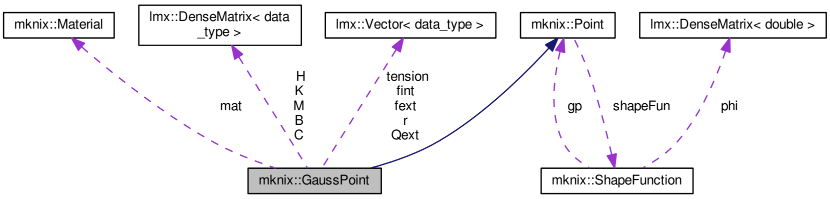
6.32 mknix::GaussPoint Class Reference

```
#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 (lmx::Matrix< data_type > &)
- void assembleHij (lmx::Matrix< data_type > &)
- void assembleQext (lmx::Vector< data_type > &)
- virtual void assembleMij (lmx::Matrix< data_type > &)=0
- virtual void assembleKij (lmx::Matrix< data_type > &)=0
- virtual void assembleRi (lmx::Vector< data_type > &, int)=0
- virtual void assembleFint (lmx::Vector< data_type > &)=0
- virtual void assembleFext (lmx::Vector< data_type > &)=0
- virtual double calcPotentialE (const lmx::Vector< data_type > &)=0
- virtual double calcKineticE (const lmx::Vector< data_type > &)=0
- virtual double calcElasticE ()=0
- void gnuplotOutStress (std::ofstream &)

Protected Attributes

- int num
- double weight
- Material ∗ mat
- bool stressPoint
- lmx::DenseMatrix< data_type > B
- lmx::DenseMatrix< data_type > C
- lmx::DenseMatrix< data_type > H
- lmx::DenseMatrix< data_type > M
- lmx::DenseMatrix< data_type > K
- lmx::Vector< data_type > tension
- lmx::Vector< data_type > r
- lmx::Vector< data_type > Qext
- lmx::Vector< data_type > fint
- lmx::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 (lmx::Matrix< data_type > & *globalCapacity*)

Definition at line 190 of file gausspoint.cpp.

6.32.3.2 virtual void mknix::GaussPoint::assembleFext (lmx::Vector< data_type > &) [pure virtual]

Implemented in mknix::GaussPoint2D, and mknix::GaussPoint3D.

6.32.3.3 virtual void mknix::GaussPoint::assembleFint (lmx::Vector< data_type > &) [pure virtual]

Implemented in mknix::GaussPoint2D, and mknix::GaussPoint3D.

6.32.3.4 void mknix::GaussPoint::assembleHij (lmx::Matrix< data_type > & *globalConductivity*)

Definition at line 202 of file gausspoint.cpp.

6.32.3.5 virtual void mknix::GaussPoint::assembleKij (lmx::Matrix< data_type > &) [pure virtual]

Implemented in mknix::GaussPoint2D, and mknix::GaussPoint3D.

6.32.3.6 virtual void mknix::GaussPoint::assembleMij (lmx::Matrix< data_type > &) [pure virtual]

Implemented in mknix::GaussPoint2D, and mknix::GaussPoint3D.

6.32.3.7 void mknix::GaussPoint::assembleQext (lmx::Vector< data_type > & *globalHeat*)

Definition at line 214 of file gausspoint.cpp.

6.32.3.8 virtual void mknix::GaussPoint::assembleRi (lmx::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 lmx::Vector< data_type > &) [pure virtual]

Implemented in mknix::GaussPoint2D, and mknix::GaussPoint3D.

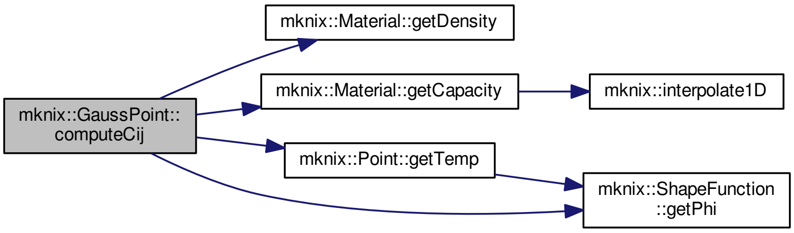
6.32.3.11 virtual double mknix::GaussPoint::calcPotentialE (const lmx::Vector< data_type > &) [pure virtual]

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.

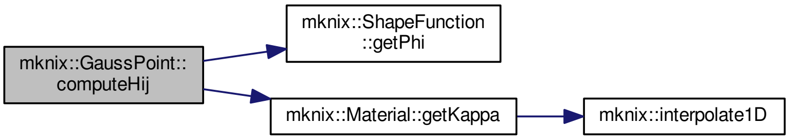
6.32.3.14 virtual void mknix::GaussPoint::computeFint ( ) [pure virtual]

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.

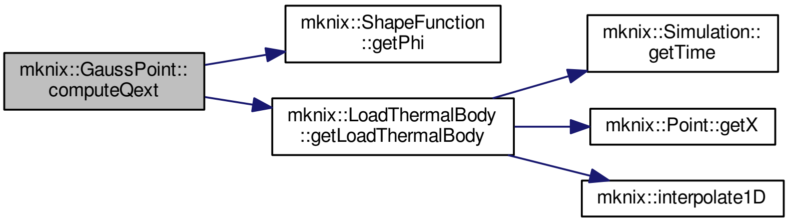
6.32.3.20 virtual void mknix::GaussPoint::computeNLStress ( ) [pure virtual]

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.

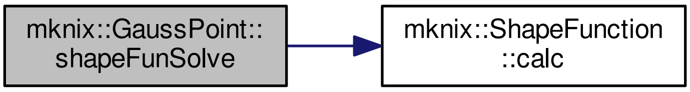
6.32.3.25 void mknix::GaussPoint::shapeFunSolve (std::string *type_in,* double *q_in*) [override], [virtual]

Reimplemented from mknix::Point.

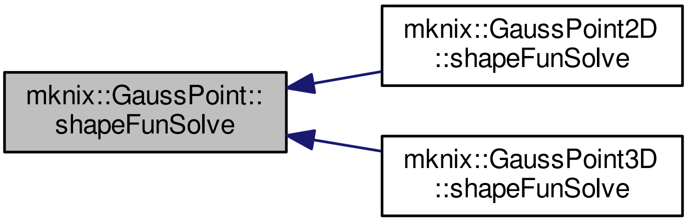
Reimplemented in mknix::GaussPoint2D, and mknix::GaussPoint3D.

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 lmx::DenseMatrix<data_type> mknix::GaussPoint::B [protected]

Definition at line 116 of file gausspoint.h.

6.32.4.2 lmx::DenseMatrix<data_type> mknix::GaussPoint::C [protected]

Definition at line 117 of file gausspoint.h.

6.32.4.3 lmx::Vector<data_type> mknix::GaussPoint::fext [protected]

Definition at line 127 of file gausspoint.h.

6.32.4.4 lmx::Vector<data_type> mknix::GaussPoint::fint [protected]

Definition at line 126 of file gausspoint.h.

6.32.4.5 lmx::DenseMatrix<data_type> mknix::GaussPoint::H [protected]

Definition at line 118 of file gausspoint.h.

6.32.4.6 lmx::DenseMatrix<data_type> mknix::GaussPoint::K [protected]

Definition at line 120 of file gausspoint.h.

6.32.4.7 lmx::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 lmx::Vector<data_type> mknix::GaussPoint::Qext [protected]

Definition at line 125 of file gausspoint.h.

6.32.4.11 lmx::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 lmx::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.

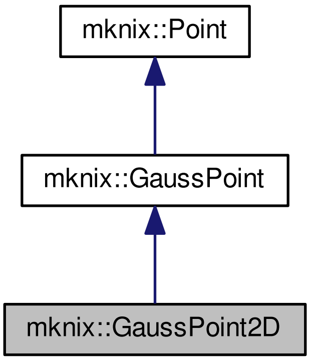
The documentation for this class was generated from the following files:

- gausspoint.h
- gausspoint.cpp

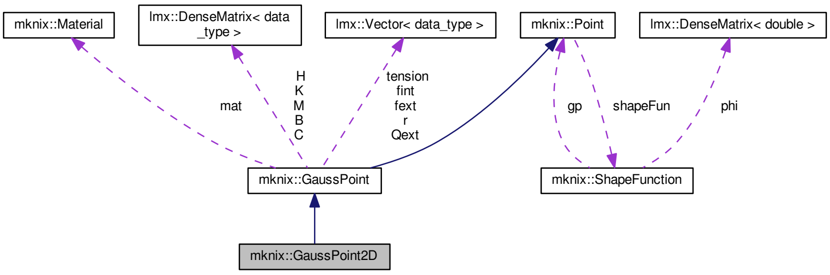
6.33 mknix::GaussPoint2D Class Reference

```
#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 (lmx::Matrix< data_type > &) override
- void assembleKij (lmx::Matrix< data_type > &) override
- void assembleRi (lmx::Vector< data_type > &, int) override
- void assembleFint (lmx::Vector< data_type > &) override
- void assembleFext (lmx::Vector< data_type > &) override
- double calcPotentialE (const lmx::Vector< data_type > &) override
- double calcKineticE (const lmx::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 (lmx::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 (lmx::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 (lmx::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 (lmx::Matrix< data_type > & *globalMass*) [override], [virtual]

Implements mknix::GaussPoint.

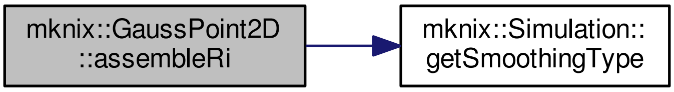
Definition at line 471 of file gausspoint2D.cpp.

6.33.3.5 void mknix::GaussPoint2D::assembleRi (lmx::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:

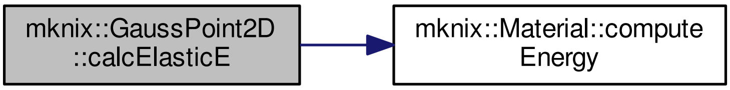

mknix::GaussPoint2D ::assembleRi
mknix::Simulation:: getSmoothingType

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:

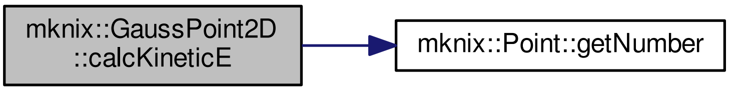

mknix::GaussPoint2D ::calcElasticE
mknix::Material::compute Energy

6.33.3.7 double mknix::GaussPoint2D::calcKineticE (const lmx::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:

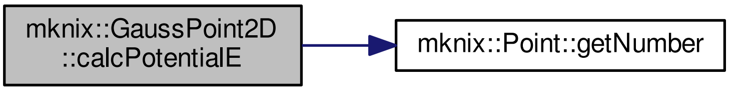


6.33.3.8 double mknix::GaussPoint2D::calcPotentialE (const lmx::Vector< data_type > & *q*) [override], [virtual]

Implements mknix::GaussPoint.

Definition at line 564 of file gausspoint2D.cpp.

Here is the call graph for this function:

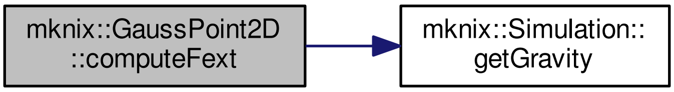


6.33.3.9 void mknix::GaussPoint2D::computeFext ( ) [override], [virtual]

Implements mknix::GaussPoint.

Definition at line 271 of file gausspoint2D.cpp.

Here is the call graph for this function:

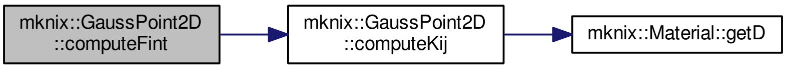

mknix::GaussPoint2D ::computeFext
mknix::Simulation:: getGravity

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:

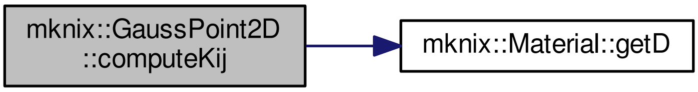

mknix::GaussPoint2D ::computeFint
mknix::GaussPoint2D ::computeKij
mknix::Material::getD

6.33.3.11 void mknix::GaussPoint2D::computeKij ( ) [override], [virtual]

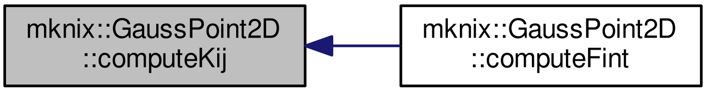
Implements mknix::GaussPoint.

Definition at line 133 of file gausspoint2D.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

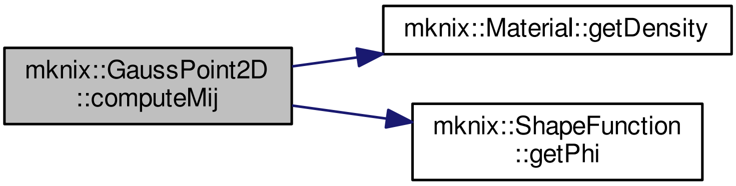

mknix::GaussPoint2D ::computeKij
mknix::GaussPoint2D ::computeFint

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:

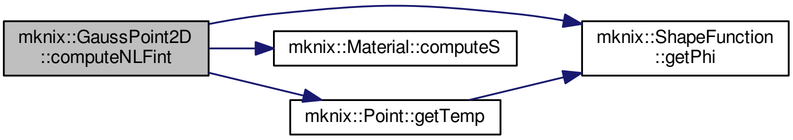

mknix::Material::getDensity
mknix::GaussPoint2D ::computeMij
mknix::ShapeFunction ::getPhi

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:

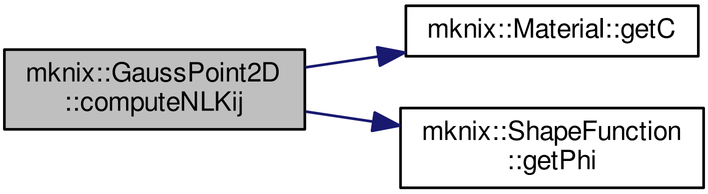

mknix::GaussPoint2D ::computeNLFint
mknix::Material::computeS
mknix::ShapeFunction ::getPhi
mknix::Point::getTemp

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:

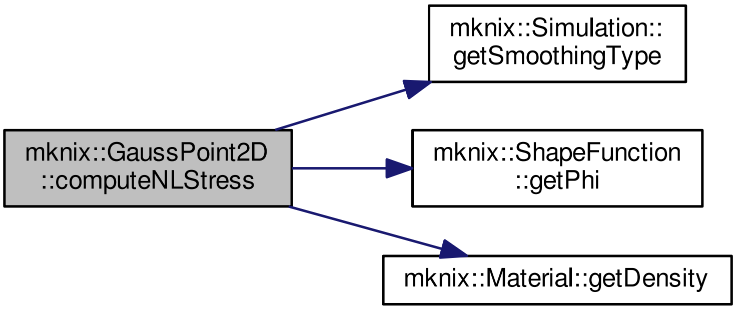

mknix::GaussPoint2D ::computeNLKij
mknix::Material::getC
mknix::ShapeFunction ::getPhi

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:

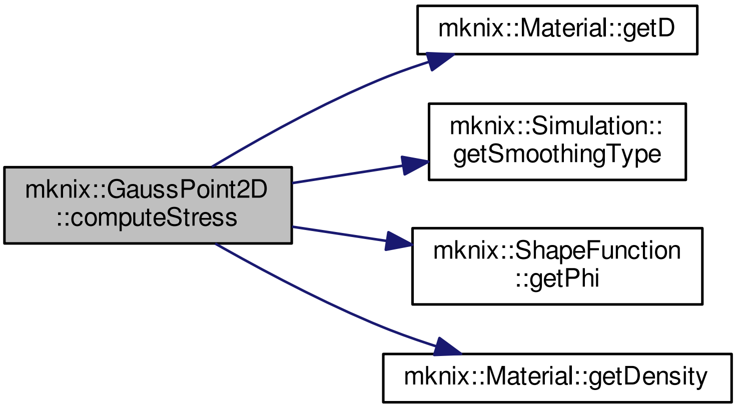


6.33.3.16 void mknix::GaussPoint2D::computeStress ( ) [override], [virtual]

Implements mknix::GaussPoint.

Definition at line 160 of file gausspoint2D.cpp.

Here is the call graph for this function:

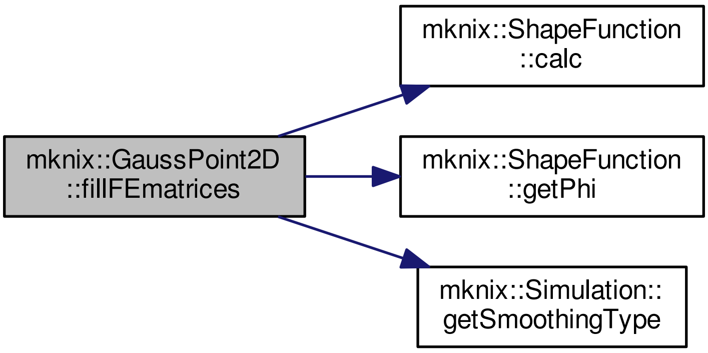


6.33.3.17 void mknix::GaussPoint2D::fillFEmatrices ( ) [override], [virtual]

Implements mknix::GaussPoint.

Definition at line 77 of file gausspoint2D.cpp.

Here is the call graph for this function:

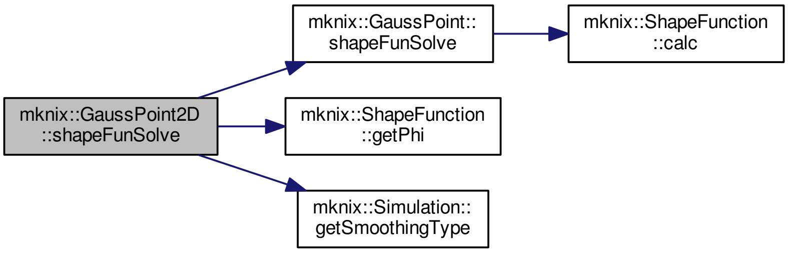

mknix::GaussPoint2D ::fillFEmatrices
mknix::ShapeFunction ::calc
mknix::ShapeFunction ::getPhi
mknix::Simulation:: getSmoothingType

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:


mknix::GaussPoint2D ::shapeFunSolve
mknix::GaussPoint:: shapeFunSolve
mknix::ShapeFunction ::calc
mknix::ShapeFunction ::getPhi
mknix::Simulation:: getSmoothingType

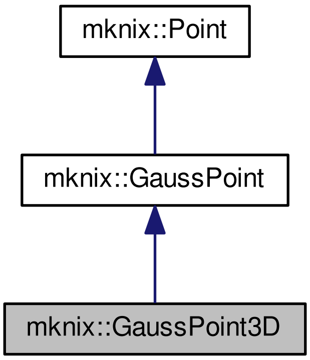
The documentation for this class was generated from the following files:

- gausspoint2D.h
- gausspoint2D.cpp

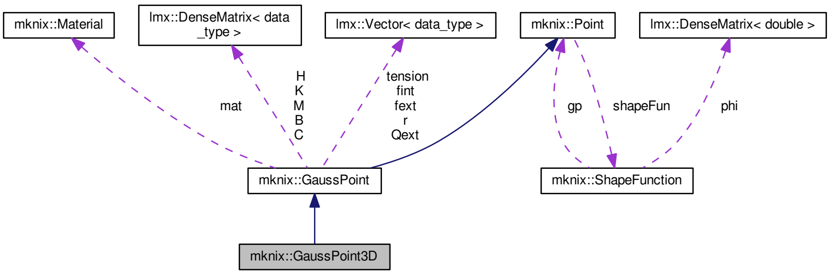
6.34 mknix::GaussPoint3D Class Reference

```
#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 (lmx::Matrix< data_type > &) override
- void assembleKij (lmx::Matrix< data_type > &) override
- void assembleRi (lmx::Vector< data_type > &, int) override

- void assembleFint (lmx::Vector< data_type > &) override
- void assembleFext (lmx::Vector< data_type > &) override
- double calcPotentialE (const lmx::Vector< data_type > &) override
- double calcKineticE (const lmx::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 (lmx::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 (lmx::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 (lmx::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 (lmx::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 (lmx::Vector< data_type > & *bodyR*, int *firstNode*) [override], [virtual]

Implements mknix::GaussPoint.

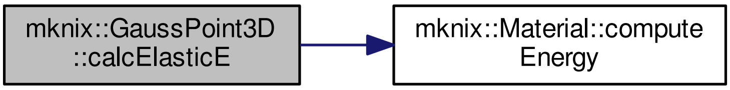
Definition at line 539 of file gausspoint3D.cpp.

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:

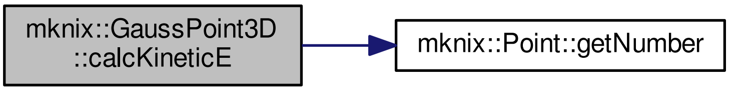

mknix::GaussPoint3D ::calcElasticE
mknix::Material::compute Energy

6.34.3.7 double mknix::GaussPoint3D::calcKineticE (const lmx::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:

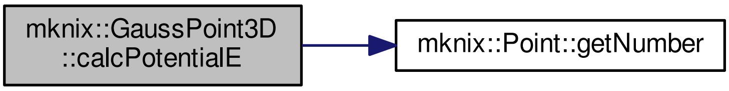


6.34.3.8 double mknix::GaussPoint3D::calcPotentialE (const lmx::Vector< data_type > & *q*) [override], [virtual]

Implements mknix::GaussPoint.

Definition at line 578 of file gausspoint3D.cpp.

Here is the call graph for this function:

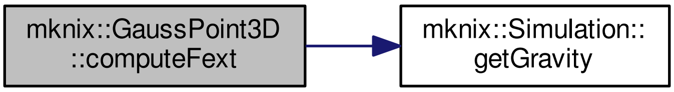


6.34.3.9 void mknix::GaussPoint3D::computeFext ( ) [override], [virtual]

Implements mknix::GaussPoint.

Definition at line 285 of file gausspoint3D.cpp.

Here is the call graph for this function:

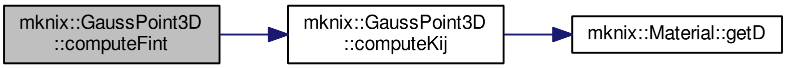

mknix::GaussPoint3D ::computeFext
mknix::Simulation:: getGravity

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:

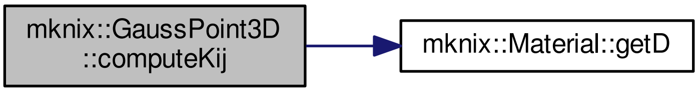

mknix::GaussPoint3D ::computeFint
mknix::GaussPoint3D ::computeKij
mknix::Material::getD

6.34.3.11 void mknix::GaussPoint3D::computeKij ( ) [override], [virtual]

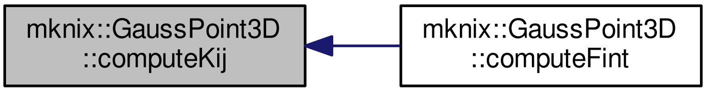
Implements mknix::GaussPoint.

Definition at line 137 of file gausspoint3D.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

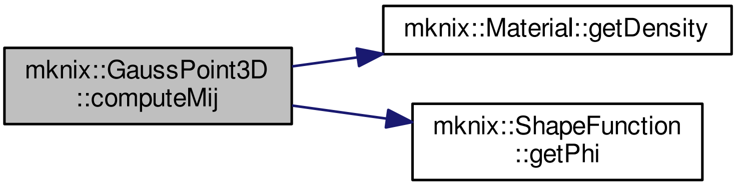

mknix::GaussPoint3D ::computeKij
mknix::GaussPoint3D ::computeFint

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:

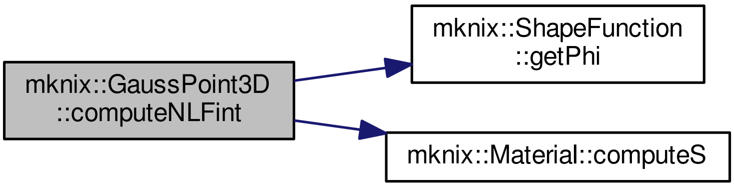

mknix::GaussPoint3D ::computeMij
mknix::Material::getDensity
mknix::ShapeFunction ::getPhi

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:

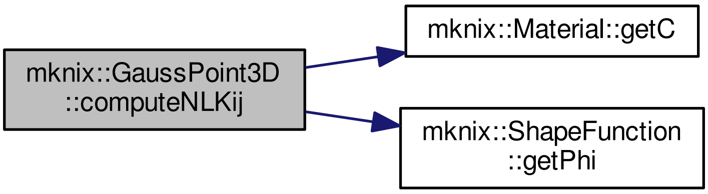

mknix::GaussPoint3D ::computeNLFint
mknix::ShapeFunction ::getPhi
mknix::Material::computeS

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:

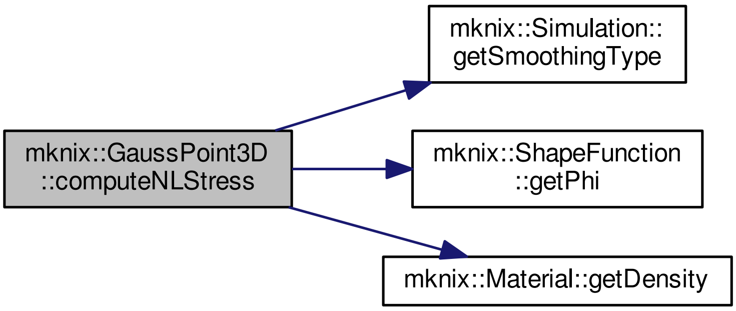

mknix::GaussPoint3D ::computeNLKij
mknix::Material::getC
mknix::ShapeFunction ::getPhi

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:

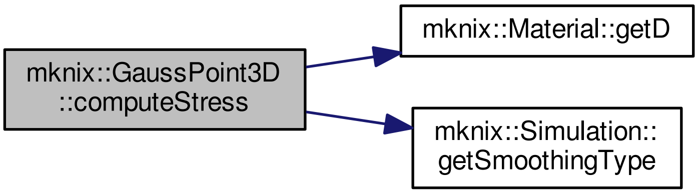

mknix::GaussPoint3D ::computeNLStress
mknix::Simulation:: getSmoothingType
mknix::ShapeFunction ::getPhi
mknix::Material::getDensity

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:

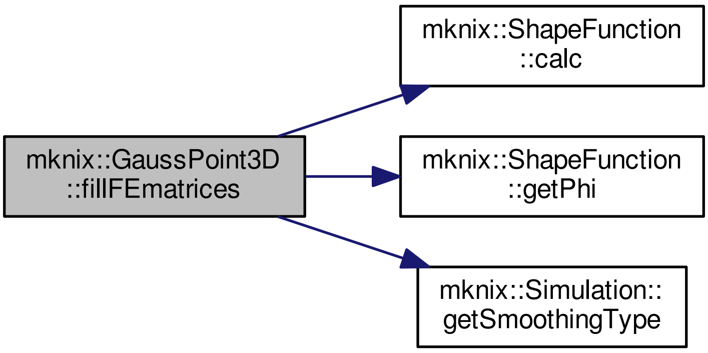

mknix::GaussPoint3D ::computeStress
mknix::Material::getD
mknix::Simulation:: getSmoothingType

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:

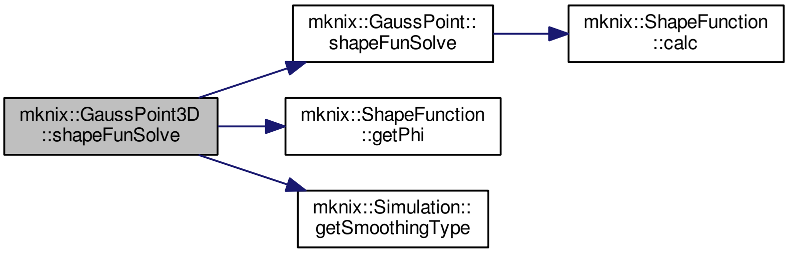

mknix::ShapeFunction ::calc
mknix::GaussPoint3D ::fillFEmatrices
mknix::ShapeFunction ::getPhi
mknix::Simulation:: getSmoothingType

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:


mknix::GaussPoint:: shapeFunSolve
mknix::ShapeFunction ::calc
mknix::GaussPoint3D ::shapeFunSolve
mknix::ShapeFunction ::getPhi
mknix::Simulation:: getSmoothingType

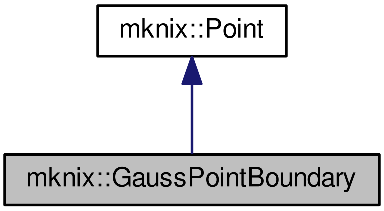
The documentation for this class was generated from the following files:

- gausspoint3D.h
- gausspoint3D.cpp

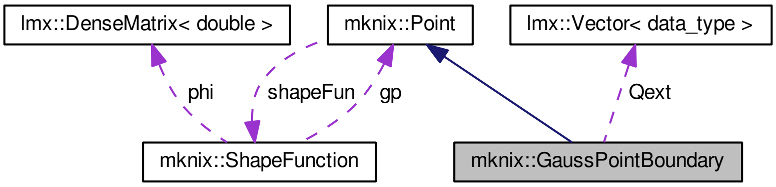
6.35 mknix::GaussPointBoundary Class Reference

```
#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 (lmx::Vector< data_type > &)

Protected Member Functions

- void initializeMatVecs ()

Protected Attributes

- int num
- double weight
- lmx::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

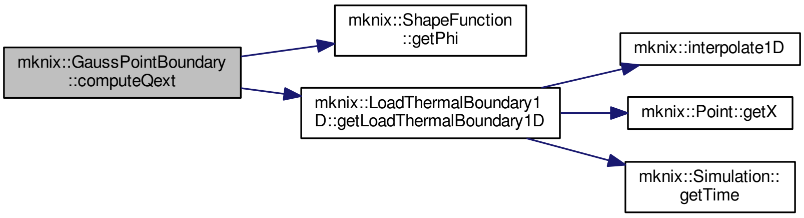
6.35.3.1 void mknix::GaussPointBoundary::assembleQext (lmx::Vector< data_type > & *globalHeat*)

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.

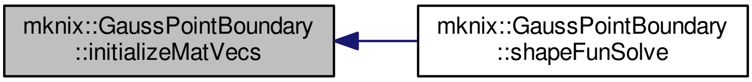
Here is the call graph for this function:


mknix::GaussPointBoundary ::computeQext
mknix::ShapeFunction ::getPhi
mknix::LoadThermalBoundary1 D::getLoadThermalBoundary1D
mknix::interpolate1D
mknix::Point::getX
mknix::Simulation:: getTime

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:

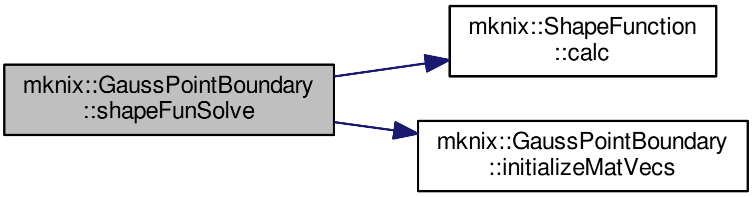

mknix::GaussPointBoundary ::initializeMatVecs
mknix::GaussPointBoundary ::shapeFunSolve

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:


mknix::GaussPointBoundary ::shapeFunSolve
mknix::ShapeFunction ::calc
mknix::GaussPointBoundary ::initializeMatVecs

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 lmx::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.

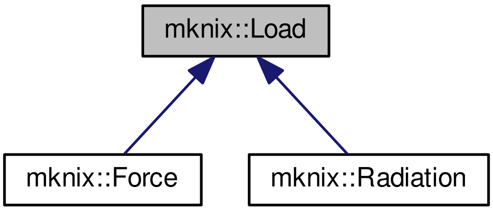
The documentation for this class was generated from the following files:

- gausspointboundary.h
- gausspointboundary.cpp

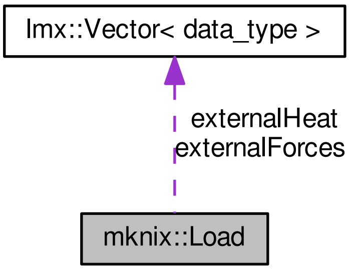
6.36 mknix::Load Class Reference

```
#include <load.h>
```

Inheritance diagram for mknix::Load:



Collaboration diagram for mknix::Load:



Public Member Functions

- Load ()
- virtual ∼Load ()
- virtual void assembleExternalForces (lmx::Vector< data_type > &)
- virtual void outputToFile (std::ofstream ∗)=0

Protected Attributes

- std::vector< Node ∗ > nodes
- lmx::Vector< data_type > externalForces
- lmx::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]

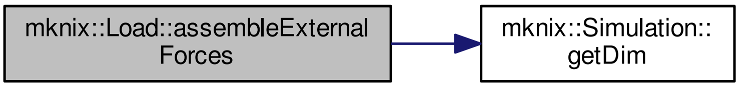
Definition at line 32 of file load.cpp.

6.36.3 Member Function Documentation

6.36.3.1 void mknix::Load::assembleExternalForces (lmx::Vector< data_type > & *globalExternalForces*) [virtual]

Definition at line 38 of file load.cpp.

Here is the call graph for this function:


mknix::Load::assembleExternal Forces
mknix::Simulation:: getDim

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 lmx::Vector<data_type> mknix::Load::externalForces [protected]

Definition at line 44 of file load.h.

6.36.4.2 lmx::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 (lmx::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 (lmx::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 ( )

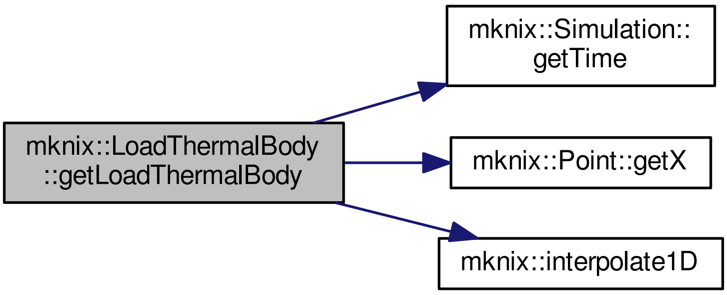
Definition at line 48 of file loadthermalbody.cpp.

6.38.3 Member Function Documentation

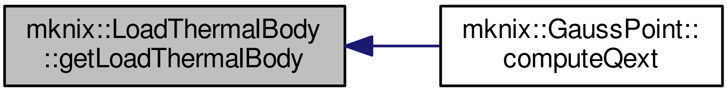
6.38.3.1 double mknix::LoadThermalBody::getLoadThermalBody (**Point** ∗ *thePoint*)

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::srim `[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 ( )

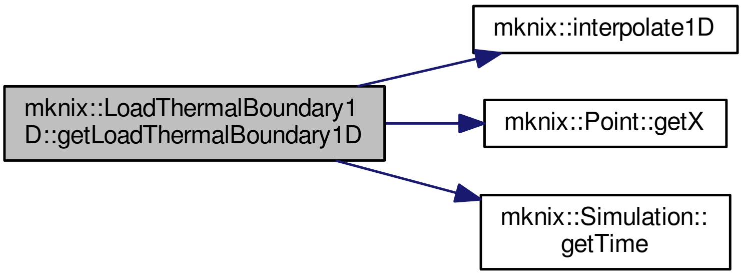
Definition at line 33 of file loadthermalboundary1D.cpp.

6.39.3 Member Function Documentation

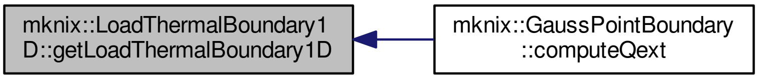
6.39.3.1 double mknix::LoadThermalBoundary1D::getLoadThermalBoundary1D (Point ∗ *thePoint*)

Definition at line 78 of file loadthermalboundary1D.cpp.

Here is the call graph for this function:


mknix::LoadThermalBoundary1D::getLoadThermalBoundary1D
mknix::interpolate1D
mknix::Point::getX
mknix::Simulation::getTime

Here is the caller graph for this function:


mknix::LoadThermalBoundary1D::getLoadThermalBoundary1D
mknix::GaussPointBoundary::computeQext

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)
- lmx::DenseMatrix< double > & getD ()
- lmx::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.

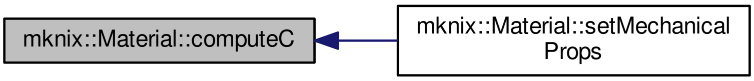
6.40.3.2 void mknix::Material::addThermalConductivity (double *temp_in,* double *conductivity_in*) [inline]

Definition at line 83 of file material.h.

6.40.3.3 void mknix::Material::computeC ( )

Definition at line 94 of file material.cpp.

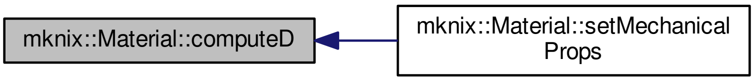
Here is the caller graph for this function:


mknix::Material::computeC
mknix::Material::setMechanical Props

6.40.3.4 void mknix::Material::computeD ( )

Definition at line 60 of file material.cpp.

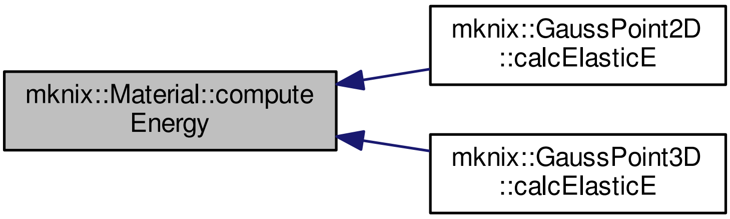
Here is the caller graph for this function:


mknix::Material::computeD
mknix::Material::setMechanical Props

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:


mknix::Material::compute Energy
mknix::GaussPoint2D ::calcElasticE
mknix::GaussPoint3D ::calcElasticE

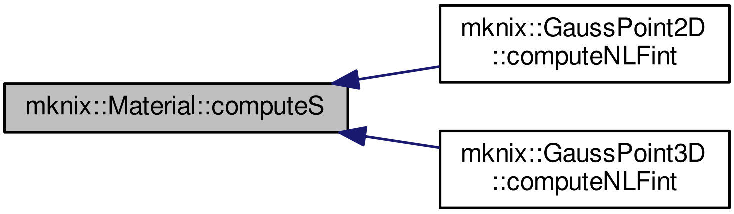
6.40.3.6 double mknix::Material::computeEnergy (const cofe::TensorRank2< 3, double > & *S*)

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:


mknix::Material::computeS
mknix::GaussPoint2D ::computeNLFint
mknix::GaussPoint3D ::computeNLFint

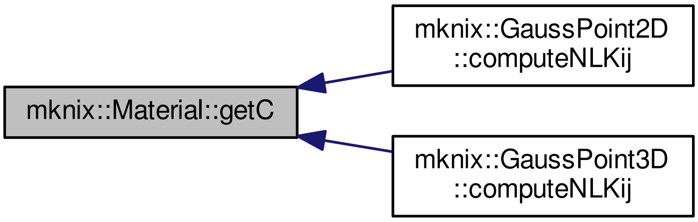
6.40.3.8 void mknix::Material::computeS (cofe::TensorRank2Sym< 3, double > & *S*, const cofe::TensorRank2< 3, double > & *F*)

Definition at line 165 of file material.cpp.

6.40.3.9 lmx::DenseMatrix<double>& mknix::Material::getC ( ) [inline]

Definition at line 74 of file material.h.

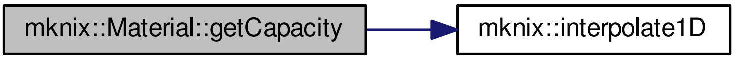
Here is the caller graph for this function:


mknix::Material::getC
mknix::GaussPoint2D ::computeNLKij
mknix::GaussPoint3D ::computeNLKij

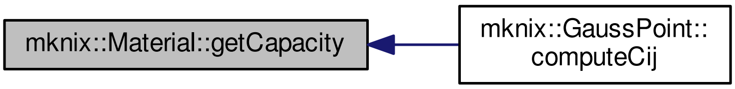
6.40.3.10 double mknix::Material::getCapacity (double *temp_in* = 0) [inline]

Definition at line 63 of file material.h.

Here is the call graph for this function:



Here is the caller graph for this function:



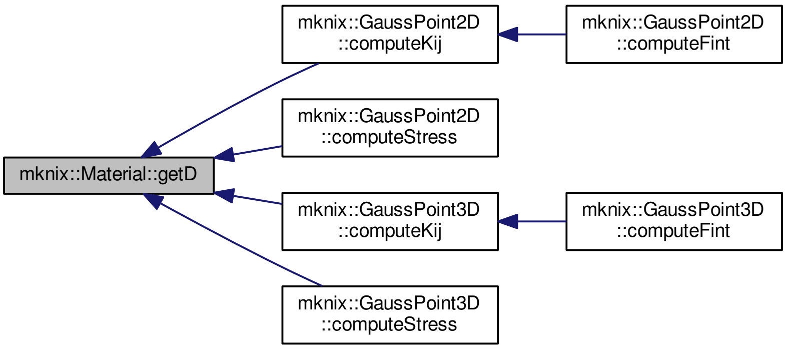
6.40.3.11 double mknix::Material::getCsym (int & *i*, int & *j*, int & *k*, int & *l*)

Definition at line 126 of file material.cpp.

6.40.3.12 lmx::DenseMatrix<double>& mknix::Material::getD ( ) `[inline]`

Definition at line 71 of file material.h.

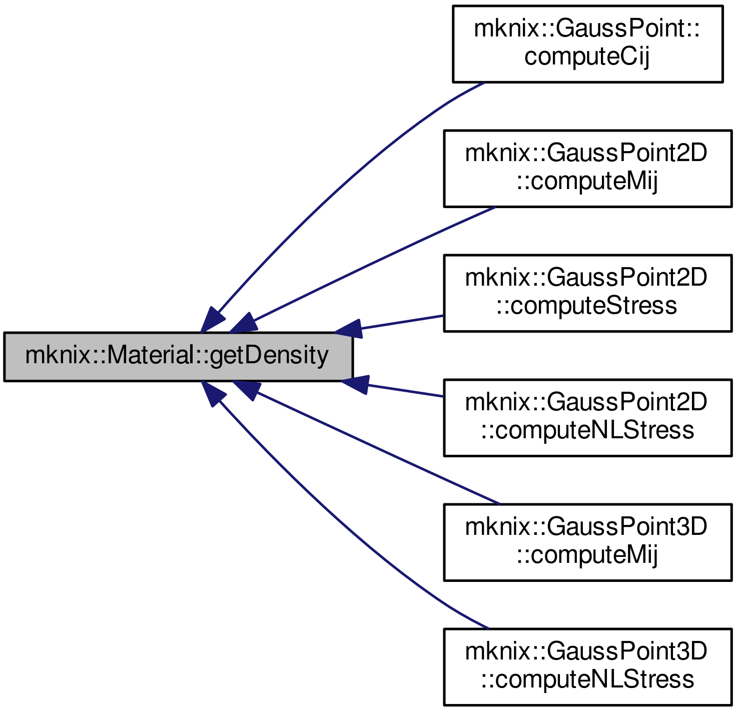
Here is the caller graph for this function:



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:



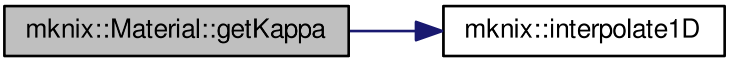
6.40.3.14 double mknix::Material::getE ( ) [inline]

Definition at line 54 of file material.h.

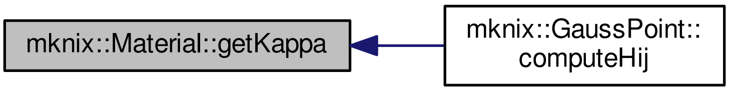
6.40.3.15 double mknix::Material::getKappa (double *temp_in* = 0) [inline]

Definition at line 67 of file material.h.

Here is the call graph for this function:



Here is the caller graph for this function:



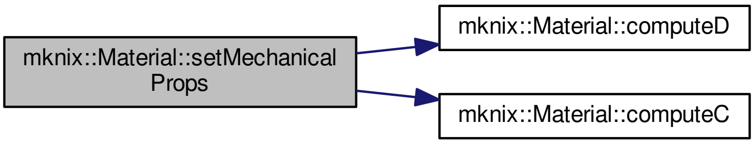
6.40.3.16 double mknix::Material::getMu ( ) `[inline]`

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:


mknix::Material::setMechanical
Props
mknix::Material::computeD
mknix::Material::computeC

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 lmx::Matrix< T > Class Template Reference

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

6.41.1 Detailed Description

template<typename T>class lmx::Matrix< T >

Definition at line 40 of file cell.h.

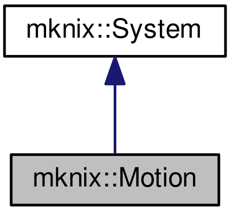
The documentation for this class was generated from the following file:

- cell.h

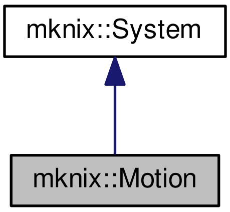
6.42 mknix::Motion Class Reference

```
#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]

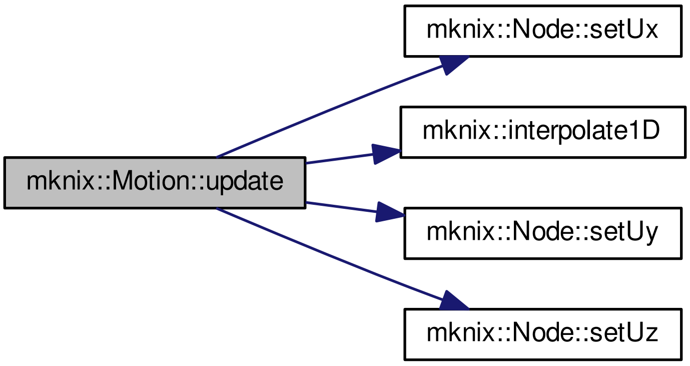
Definition at line 58 of file motion.h.

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:



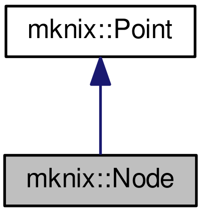
The documentation for this class was generated from the following files:

- motion.h
- motion.cpp

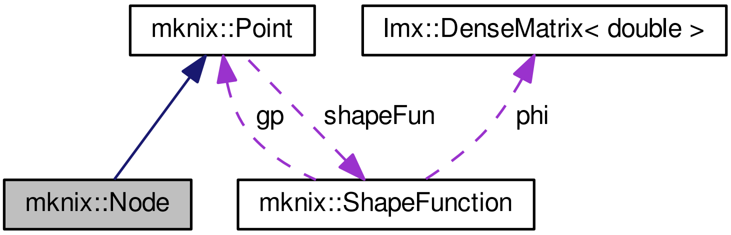
6.43 mknix::Node Class Reference

```
#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]

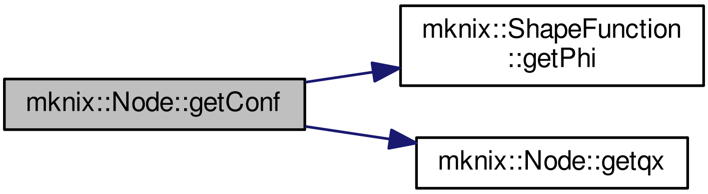
Definition at line 43 of file node.h.

6.43.3.2 double mknix::Node::getConf (int *dof*) const [override], [virtual]

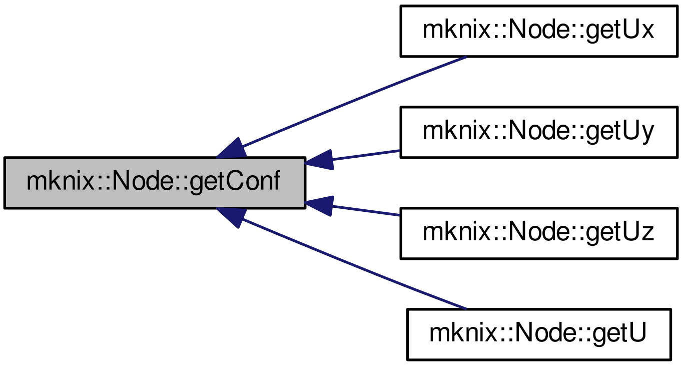
Reimplemented from mknix::Point.

Definition at line 57 of file node.cpp.

Here is the call graph for this function:



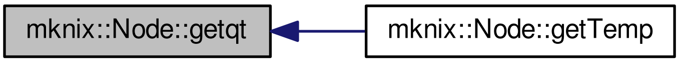
Here is the caller graph for this function:



6.43.3.3 const double& mknix::Node::getqt ( ) const [inline]

Definition at line 75 of file node.h.

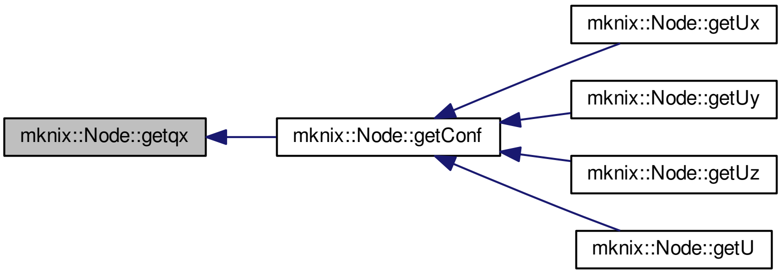
Here is the caller graph for this function:



6.43.3.4 double mknix::Node::getqx (int *i*) const [inline]

Definition at line 79 of file node.h.

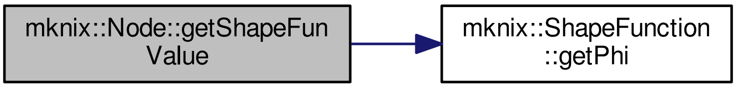
Here is the caller graph for this function:



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:

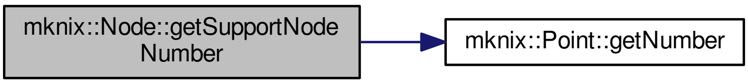


6.43.3.6 int mknix::Node::getSupportNodeNumber (int *deriv,* int *s_node*) [virtual]

Reimplemented from mknix::Point.

Definition at line 127 of file node.cpp.

Here is the call graph for this function:

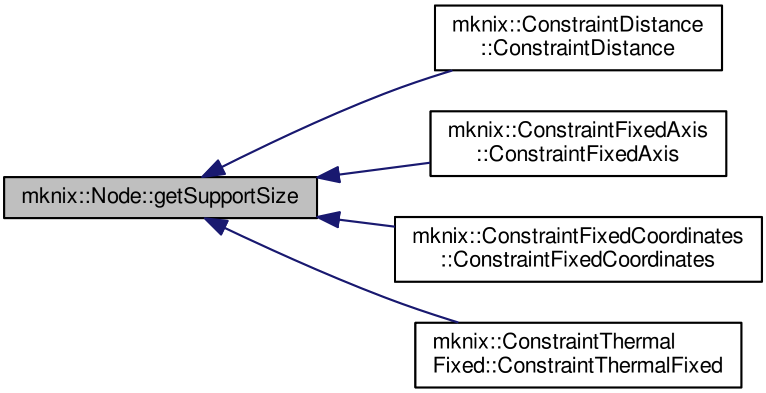


6.43.3.7 size_t mknix::Node::getSupportSize (int *deriv*) [virtual]

Reimplemented from mknix::Point.

Definition at line 100 of file node.cpp.

Here is the caller graph for this function:

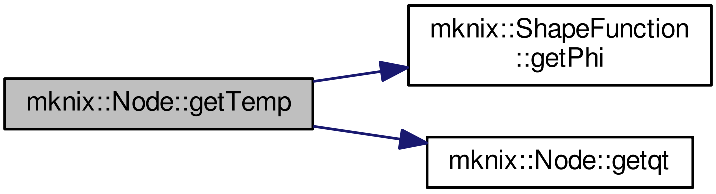

mknix::Node::getSupportSize
mknix::ConstraintDistance ::ConstraintDistance
mknix::ConstraintFixedAxis ::ConstraintFixedAxis
mknix::ConstraintFixedCoordinates ::ConstraintFixedCoordinates
mknix::ConstraintThermal Fixed::ConstraintThermalFixed

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:



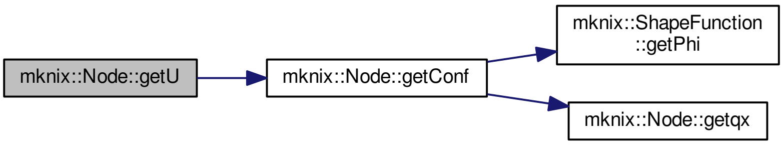
6.43.3.9 const int& mknix::Node::getThermalNumber ( ) const [inline]

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.

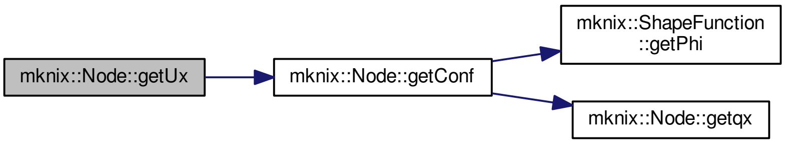
Here is the call graph for this function:



6.43.3.11 double mknix::Node::getUx ( ) const [inline]

Definition at line 87 of file node.h.

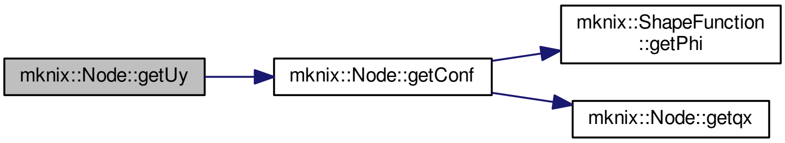
Here is the call graph for this function:



6.43.3.12 double mknix::Node::getUy ( ) const [inline]

Definition at line 91 of file node.h.

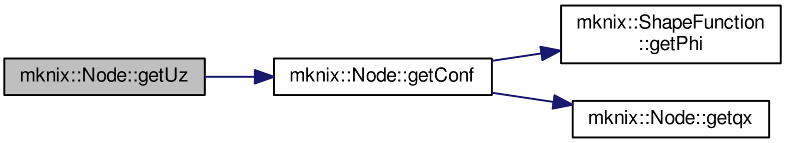
Here is the call graph for this function:



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 lmx::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 lmx::Vector< data_type > & *globalConf,* int *dim*)

Definition at line 168 of file node.cpp.

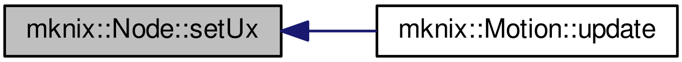
6.43.3.19 const void mknix::Node::setThermalNumber (int *num_in*) [inline]

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.

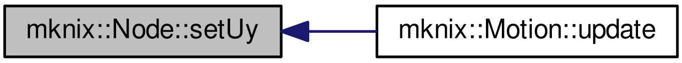
Here is the caller graph for this function:



6.43.3.21 void mknix::Node::setUy (double *uy_in*) [inline]

Definition at line 122 of file node.h.

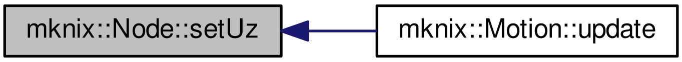
Here is the caller graph for this function:



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.

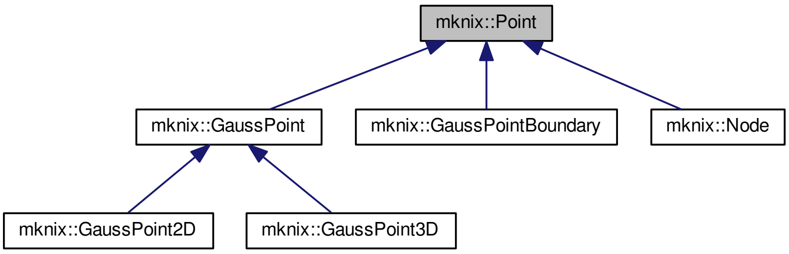
The documentation for this class was generated from the following files:

- node.h
- node.cpp

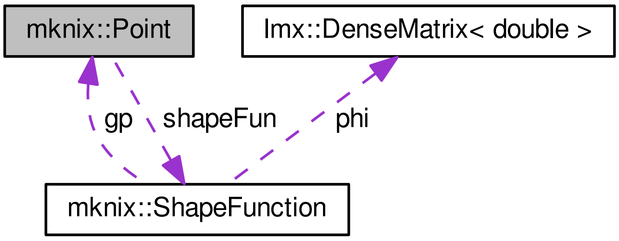
6.44 mknix::Point Class Reference

```
#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 mknix::Point::Point ( )

Definition at line 15 of file point.cpp.

6.44.2.2 mknix::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

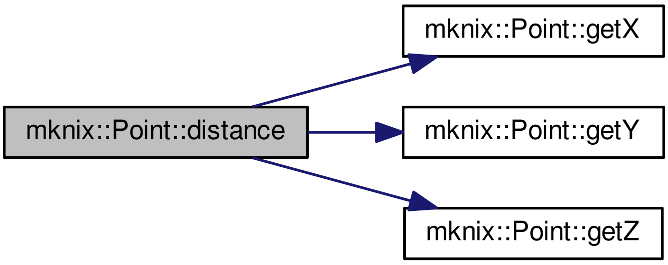
6.44.3.1 void mknix::Point::addSupportNode (Node ∗ *node_in*)

Definition at line 148 of file point.cpp.

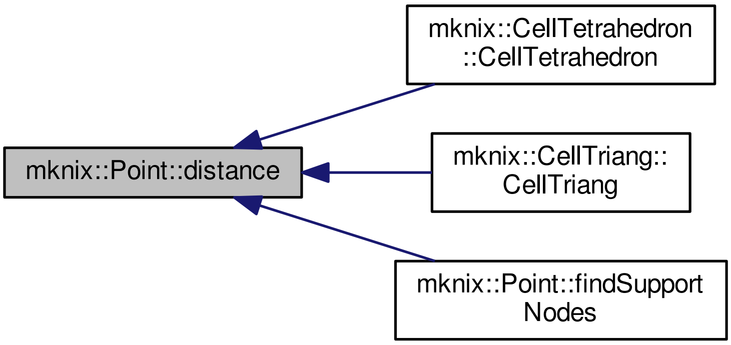
6.44.3.2 double mknix::Point::distance (Point & *node_in*) const

Definition at line 135 of file point.cpp.

Here is the call graph for this function:



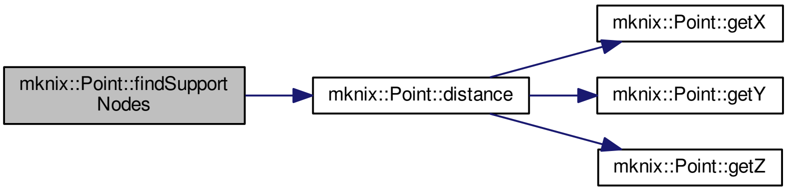
Here is the caller graph for this function:


mknix::CellTetrahedron ::CellTetrahedron
mknix::Point::distance
mknix::CellTriang:: CellTriang
mknix::Point::findSupport Nodes

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*)

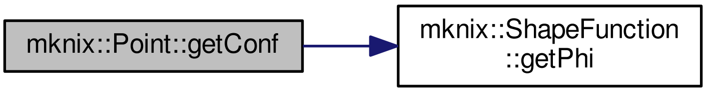
Definition at line 216 of file point.cpp.

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:



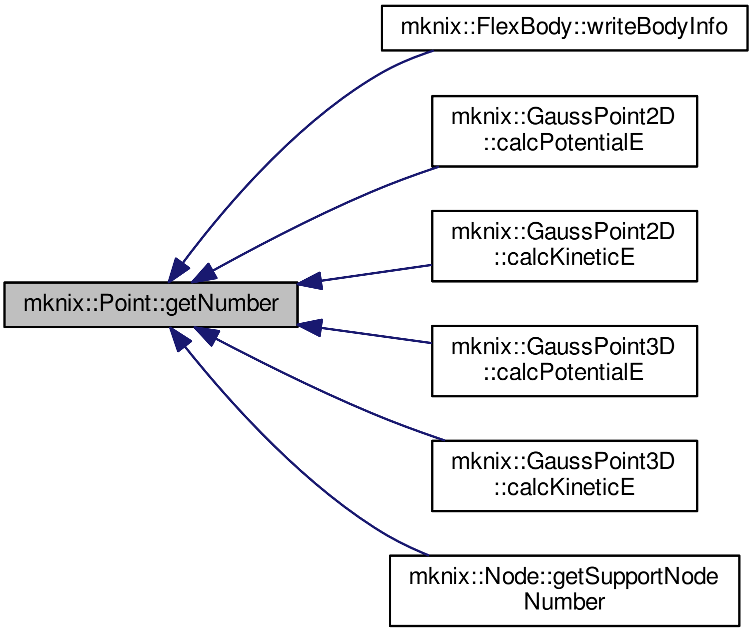
6.44.3.6 const int& mknix::Point::getDim ( ) const [inline]

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:


mknix::FlexBody::writeBodyInfo
mknix::GaussPoint2D ::calcPotentialE
mknix::GaussPoint2D ::calcKineticE
mknix::Point::getNumber
mknix::GaussPoint3D ::calcPotentialE
mknix::GaussPoint3D ::calcKineticE
mknix::Node::getSupportNode Number

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]

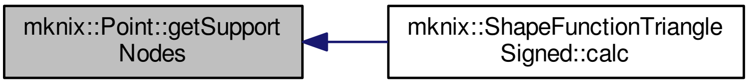
Reimplemented in mknix::Node.

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:


mknix::Point::getSupport Nodes
mknix::ShapeFunctionTriangle Signed::calc

6.44.3.11 virtual size_t mknix::Point::getSupportSize (int *deriv* = 0) [inline], [virtual]

Reimplemented in mknix::Node.

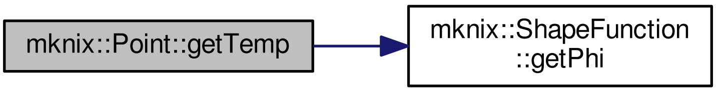
Definition at line 133 of file point.h.

6.44.3.12 double mknix::Point::getTemp ( ) const [virtual]

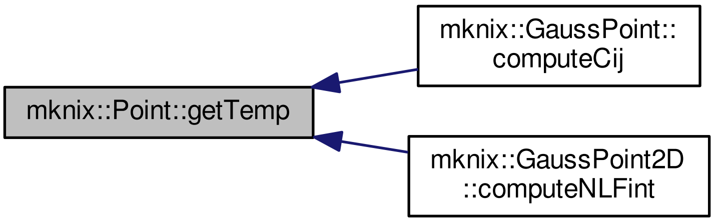
Reimplemented in mknix::Node.

Definition at line 117 of file point.cpp.

Here is the call graph for this function:



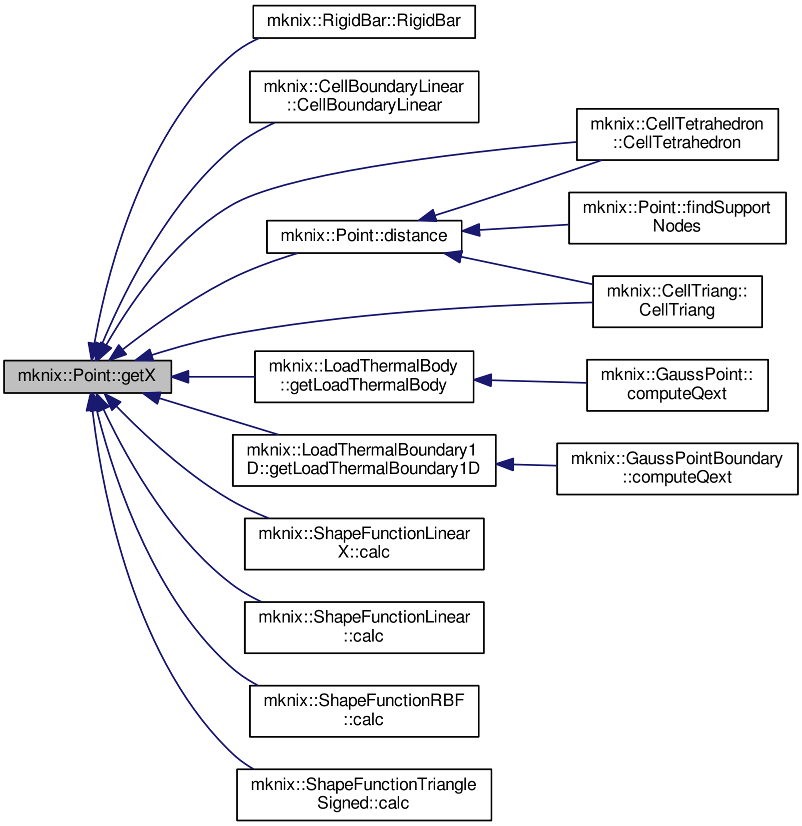
Here is the caller graph for this function:


mknix::GaussPoint:: computeCij
mknix::Point::getTemp
mknix::GaussPoint2D ::computeNLFint

6.44.3.13 const double& mknix::Point::getX ( ) const [inline]

Definition at line 97 of file point.h.

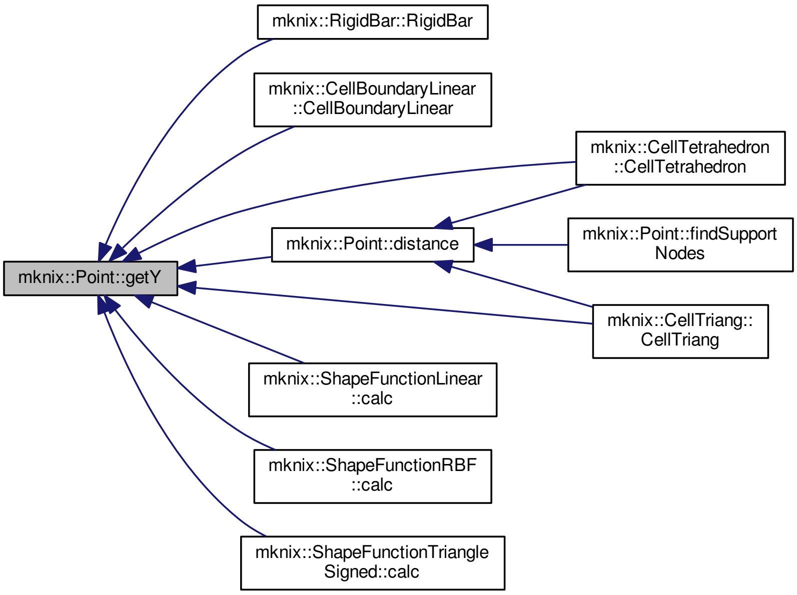
Here is the caller graph for this function:


mknix::RigidBar::RigidBar
mknix::CellBoundaryLinear ::CellBoundaryLinear
mknix::CellTetrahedron ::CellTetrahedron
mknix::Point::findSupport Nodes
mknix::Point::distance
mknix::CellTriang:: CellTriang
mknix::Point::getX
mknix::LoadThermalBody ::getLoadThermalBody
mknix::GaussPoint:: computeQext
mknix::LoadThermalBoundary1 D::getLoadThermalBoundary1D
mknix::GaussPointBoundary ::computeQext
mknix::ShapeFunctionLinear X::calc
mknix::ShapeFunctionLinear ::calc
mknix::ShapeFunctionRBF ::calc
mknix::ShapeFunctionTriangle Signed::calc

6.44.3.14 const double& mknix::Point::getY ( ) const [inline]

Definition at line 102 of file point.h.

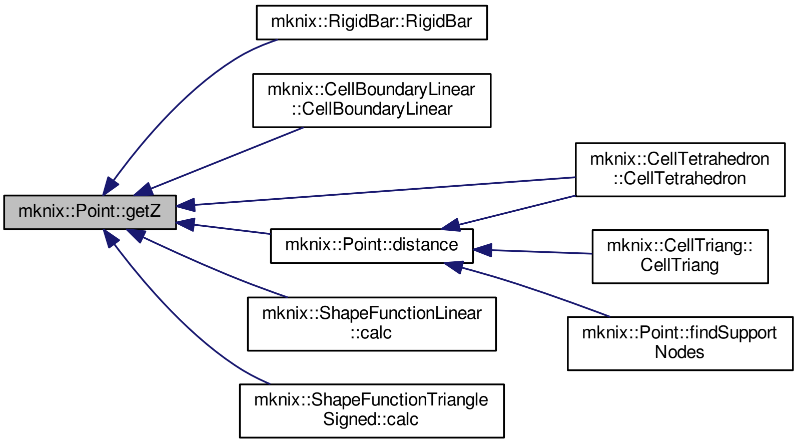
Here is the caller graph for this function:


mknix::RigidBar::RigidBar
mknix::CellBoundaryLinear ::CellBoundaryLinear
mknix::CellTetrahedron ::CellTetrahedron
mknix::Point::distance
mknix::Point::findSupport Nodes
mknix::Point::getY
mknix::CellTriang:: CellTriang
mknix::ShapeFunctionLinear ::calc
mknix::ShapeFunctionRBF ::calc
mknix::ShapeFunctionTriangle Signed::calc

6.44.3.15 const double& mknix::Point::getZ ( ) const [inline]

Definition at line 107 of file point.h.

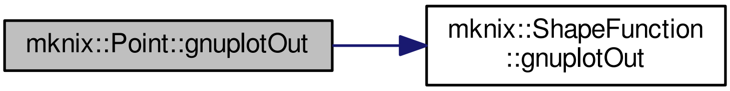
Here is the caller graph for this function:


mknix::RigidBar::RigidBar
mknix::CellBoundaryLinear ::CellBoundaryLinear
mknix::CellTetrahedron ::CellTetrahedron
mknix::Point::getZ
mknix::Point::distance
mknix::CellTriang:: CellTriang
mknix::ShapeFunctionLinear ::calc
mknix::Point::findSupport Nodes
mknix::ShapeFunctionTriangle Signed::calc

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]

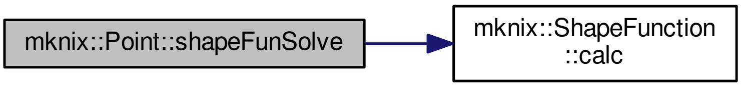
Definition at line 123 of file point.h.

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.

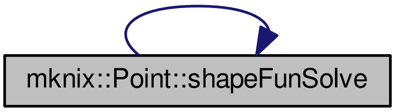
Here is the call graph for this function:



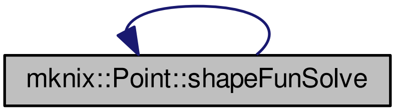
6.44.3.22 void mknix::Point::shapeFunSolve (double *q_in*) [inline]

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::alphai [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.

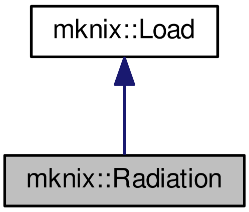
The documentation for this class was generated from the following files:

- point.h
- point.cpp

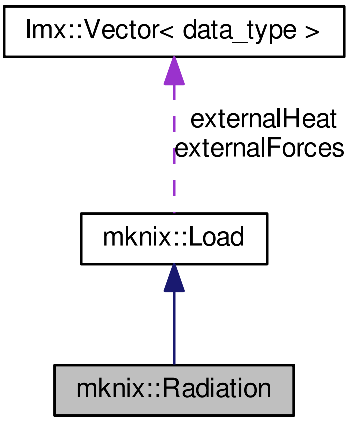
6.45 mknix::Radiation Class Reference

```
#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 ( )

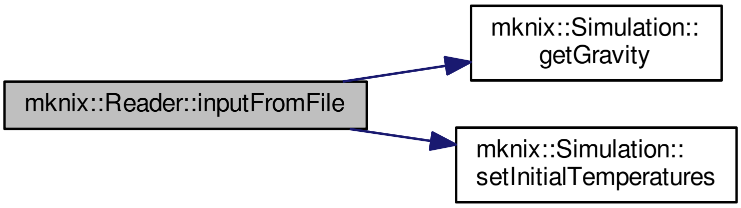
Definition at line 68 of file reader.cpp.

6.46.3 Member Function Documentation

6.46.3.1 void mknix::Reader::inputFromFile (const std::string & *fileIn*)

Definition at line 75 of file reader.cpp.

Here is the call graph for this function:


mknix::Reader::inputFromFile
mknix::Simulation::getGravity
mknix::Simulation::setInitialTemperatures

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 & 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 ( )

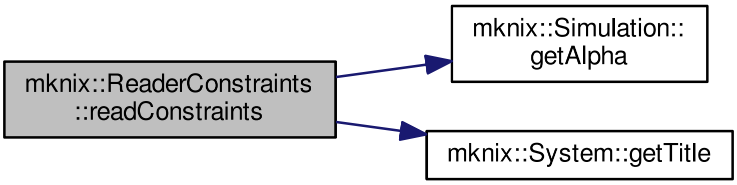
Definition at line 59 of file readerconstraints.cpp.

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:


mknix::ReaderConstraints ::readConstraints
mknix::Simulation:: getAlpha
mknix::System::getTitle

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 ( )

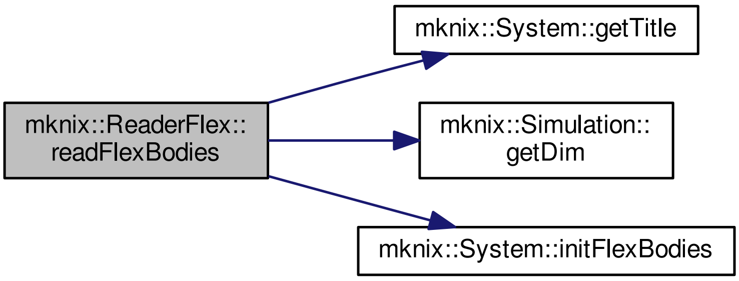
Definition at line 53 of file readerflex.cpp.

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 ( )

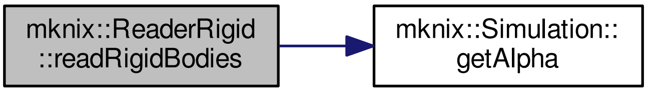
Definition at line 51 of file readerrigid.cpp.

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:


mknix::ReaderRigid ::readRigidBodies
mknix::Simulation:: getAlpha

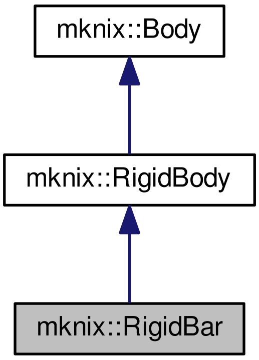
The documentation for this class was generated from the following files:

- readerrigid.h
- readerrigid.cpp

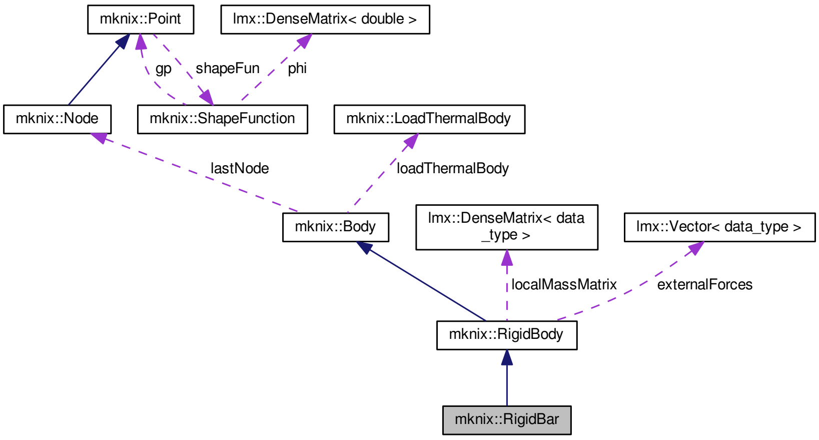
6.50 mknix::RigidBar Class Reference

```
#include <bodyrigid1D.h>
```

Inheritance diagram for mknix::RigidBar:



Collaboration diagram for mknix::RigidBar:



Public Member Functions

- RigidBar ()
- RigidBar (std::string, Node ∗, Node ∗, double)
- ∼RigidBar ()
- 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

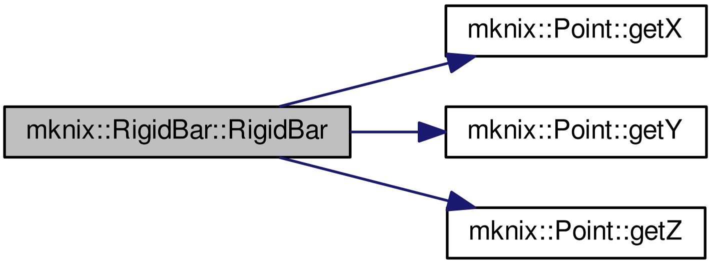
6.50.2.1 mknix::RigidBar::RigidBar ( )

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.

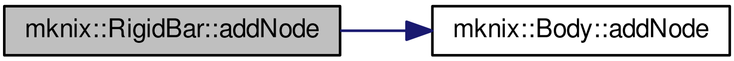
6.50.3 Member Function Documentation

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:

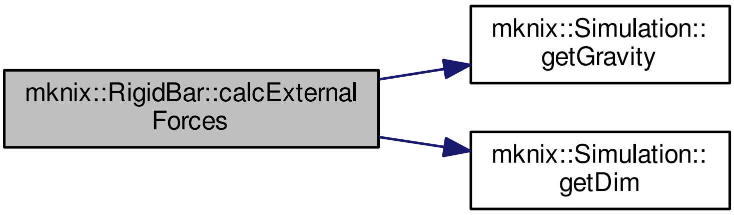


6.50.3.2 void mknix::RigidBar::calcExternalForces ( ) [virtual]

Implements mknix::Body.

Definition at line 95 of file bodyrigid1D.cpp.

Here is the call graph for this function:

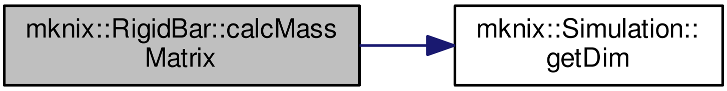

mknix::RigidBar::calcExternal Forces
mknix::Simulation:: getGravity
mknix::Simulation:: getDim

6.50.3.3 void mknix::RigidBar::calcMassMatrix ( ) [virtual]

Implements mknix::Body.

Definition at line 75 of file bodyrigid1D.cpp.

Here is the call graph for this function:



6.50.3.4 std::string mknix::RigidBar::getType ( ) [inline], [virtual]

Implements mknix::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.

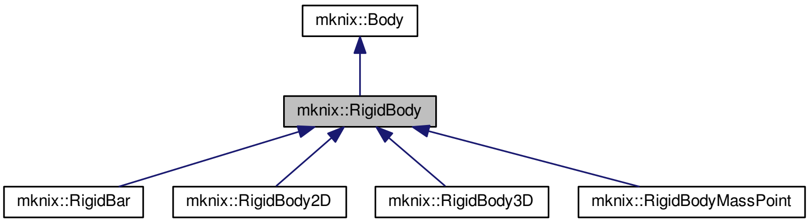
The documentation for this class was generated from the following files:

- bodyrigid1D.h
- bodyrigid1D.cpp

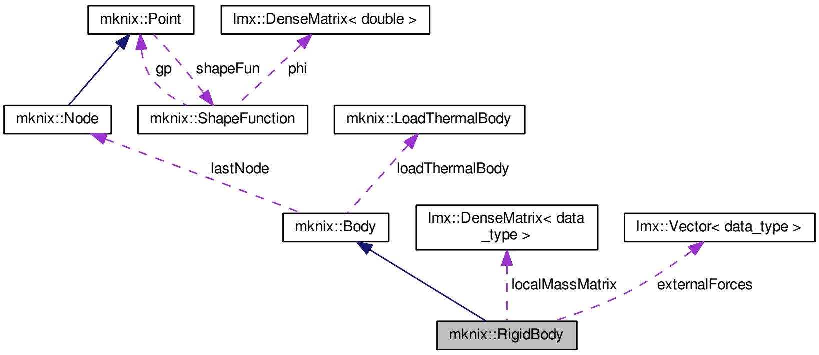
6.51 mknix::RigidBody Class Reference

`#include <bodyrigid.h>`

Inheritance diagram for mknix::RigidBody:



Collaboration diagram for mknix::RigidBody:



Public Member Functions

- RigidBody ()
- RigidBody (std::string)
- virtual ∼RigidBody ()
- void assembleMassMatrix (lmx::Matrix< data_type > &)
- void assembleExternalForces (lmx::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 lmx::Vector< data_type > &, const lmx::Vector< data_type > &)
- void outputStep (const lmx::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< lmx::Vector< data_type > ∗ > domainConf
- lmx::Vector< data_type > externalForces
- lmx::DenseMatrix< data_type > localMassMatrix
- std::vector< lmx::Vector< data_type > ∗ > energy

6.51.1 Detailed Description

Author

AUTHORS <MAILS>

Definition at line 33 of file bodyrigid.h.

6.51.2 Constructor & 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.

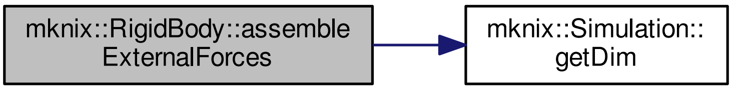
6.51.3 Member Function Documentation

6.51.3.1 void mknix::RigidBody::assembleExternalForces (lmx::Vector< data_type > & *globalExternalForces*) [virtual]

Implements mknix::Body.

Definition at line 71 of file bodyrigid.cpp.

Here is the call graph for this function:

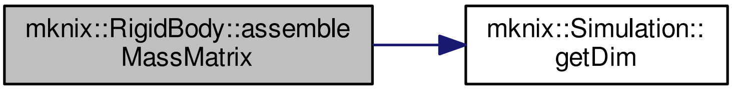


6.51.3.2 void mknix::RigidBody::assembleMassMatrix (lmx::Matrix< data_type > & *globalMass*) [virtual]

Implements mknix::Body.

Definition at line 49 of file bodyrigid.cpp.

Here is the call graph for this function:


mknix::RigidBody::assemble MassMatrix
mknix::Simulation:: getDim

6.51.3.3 Node $*$ mknix::RigidBody::getNode (int *node_number*) [virtual]

Reimplemented from mknix::Body.

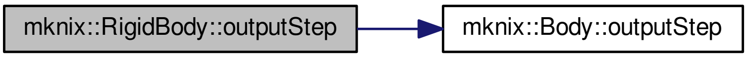
Definition at line 97 of file bodyrigid.cpp.

6.51.3.4 void mknix::RigidBody::outputStep (const lmx::Vector$<$ data_type $>$ & *q*, const lmx::Vector$<$ data_type $>$ & *qdot*) [virtual]

Implements mknix::Body.

Definition at line 122 of file bodyrigid.cpp.

Here is the call graph for this function:

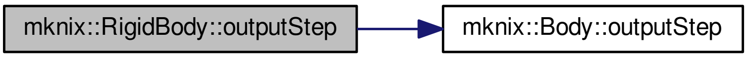


6.51.3.5 void mknix::RigidBody::outputStep (const lmx::Vector$<$ data_type $>$ & *q*) [virtual]

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

outFile	Output files

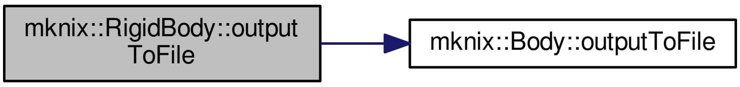
Returns

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< double > &) [pure virtual]

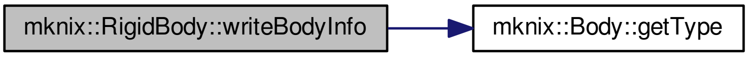
Implemented in mknix::RigidBar, mknix::RigidBody2D, mknix::RigidBody3D, and mknix::RigidBodyMassPoint.

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< lmx::Vector<data_type>∗ > mknix::RigidBody::domainConf [protected]

Definition at line 78 of file bodyrigid.h.

6.51.4.5 std::vector< lmx::Vector<data_type>∗ > mknix::RigidBody::energy [protected]

Definition at line 81 of file bodyrigid.h.

6.51.4.6 lmx::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 lmx::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.

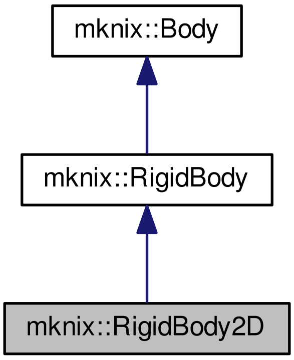
The documentation for this class was generated from the following files:

- bodyrigid.h
- bodyrigid.cpp

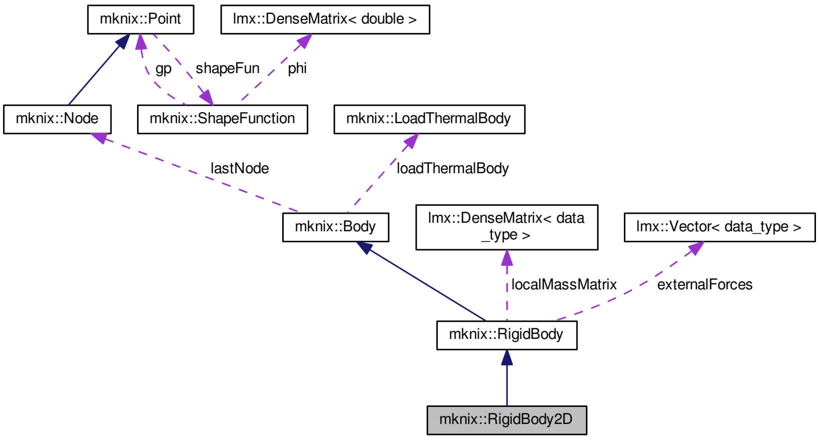
6.52 mknix::RigidBody2D Class Reference

```
#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.

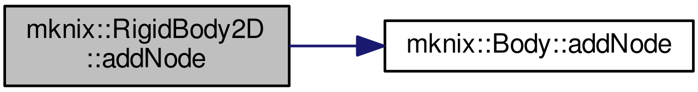
6.52.3 Member Function Documentation

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:

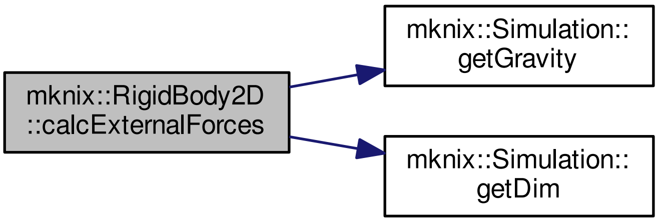


6.52.3.2 void mknix::RigidBody2D::calcExternalForces ( ) `[virtual]`

Implements mknix::Body.

Definition at line 124 of file bodyrigid2D.cpp.

Here is the call graph for this function:

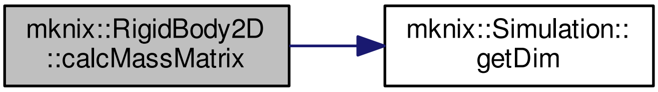

mknix::RigidBody2D ::calcExternalForces
mknix::Simulation:: getGravity
mknix::Simulation:: getDim

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:


mknix::RigidBody2D ::calcMassMatrix
mknix::Simulation:: getDim

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.

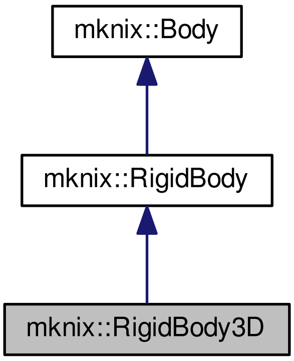
The documentation for this class was generated from the following files:

- bodyrigid2D.h
- bodyrigid2D.cpp

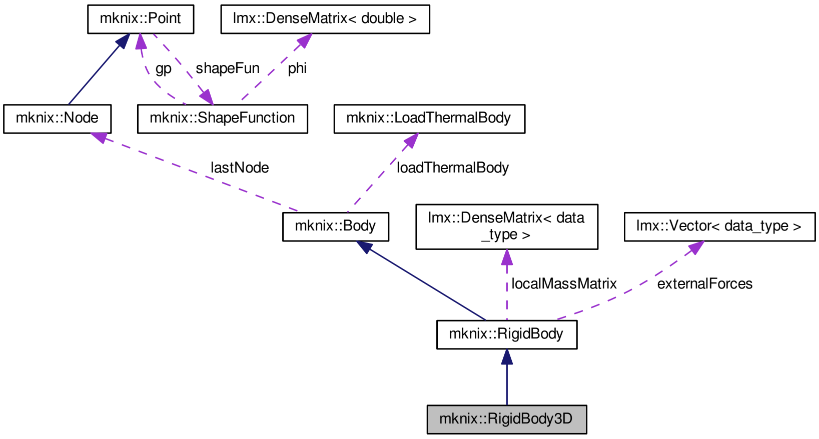
6.53 mknix::RigidBody3D Class Reference

```
#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

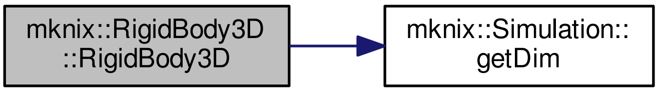
6.53.2.1 mknix::RigidBody3D::RigidBody3D ( )

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.

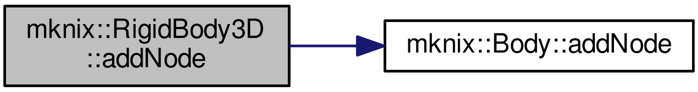
6.53.3 Member Function Documentation

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:

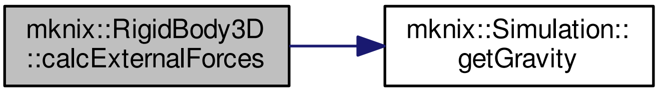


6.53.3.2 void mknix::RigidBody3D::calcExternalForces ( ) [virtual]

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.

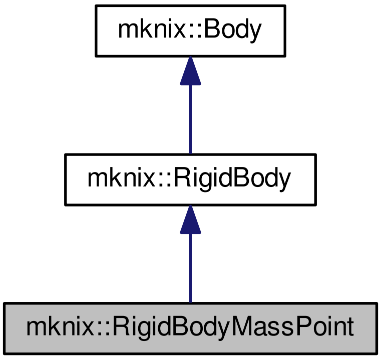
The documentation for this class was generated from the following files:

- bodyrigid3D.h
- bodyrigid3D.cpp

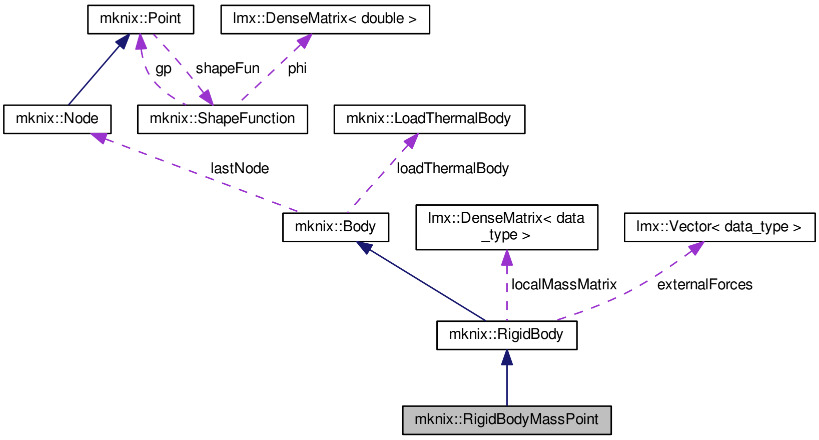
6.54 mknix::RigidBodyMassPoint Class Reference

```
#include <bodyrigid0D.h>
```

Inheritance diagram for mknix::RigidBodyMassPoint:



Collaboration diagram for mknix::RigidBodyMassPoint:



Public Member Functions

- RigidBodyMassPoint ()
- RigidBodyMassPoint (std::string, Node $*$, double)
- $\sim$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.

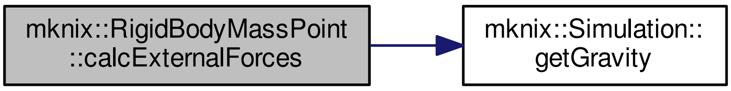
6.54.3 Member Function Documentation

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:


mknix::RigidBodyMassPoint ::calcExternalForces
mknix::Simulation:: getGravity

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.

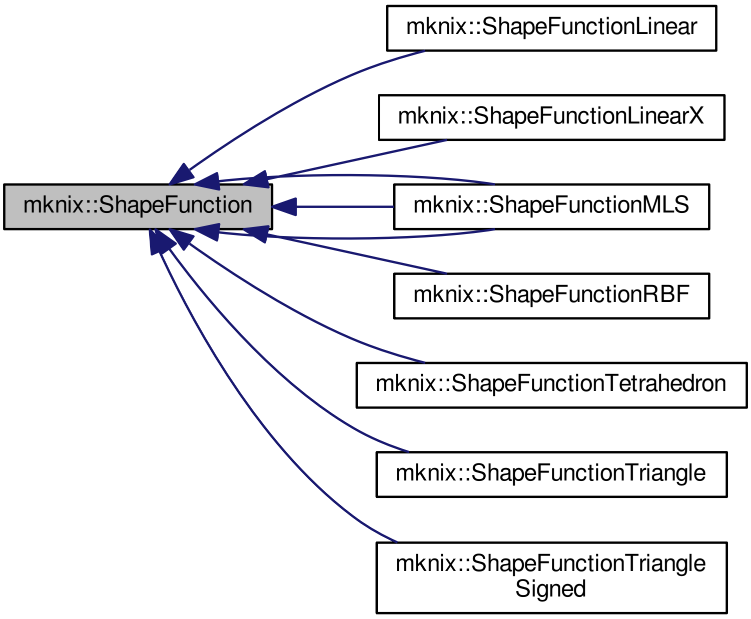
The documentation for this class was generated from the following files:

- bodyrigid0D.h
- bodyrigid0D.cpp

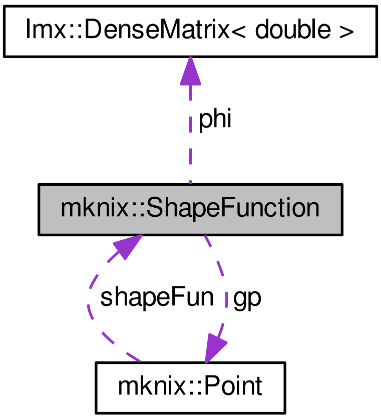
6.55 mknix::ShapeFunction Class Reference

```
#include <shapefunction.h>
```

Inheritance diagram for mknix::ShapeFunction:


mknix::ShapeFunction
mknix::ShapeFunctionLinear
mknix::ShapeFunctionLinearX
mknix::ShapeFunctionMLS
mknix::ShapeFunctionRBF
mknix::ShapeFunctionTetrahedron
mknix::ShapeFunctionTriangle
mknix::ShapeFunctionTriangle Signed

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
- lmx::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]

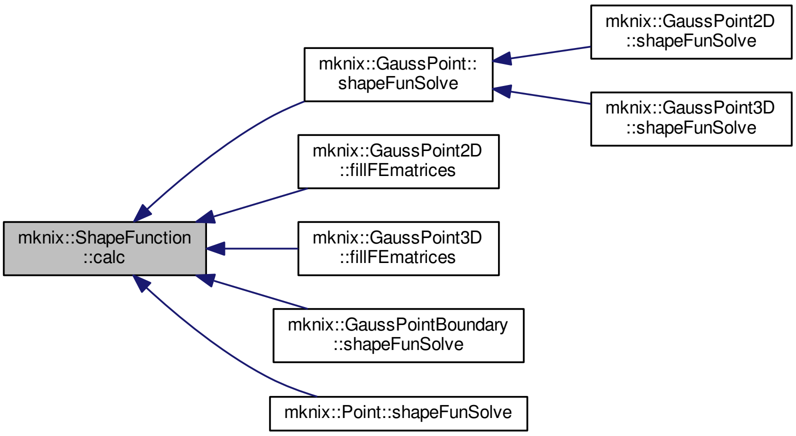
Definition at line 29 of file shapefunction.cpp.

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.

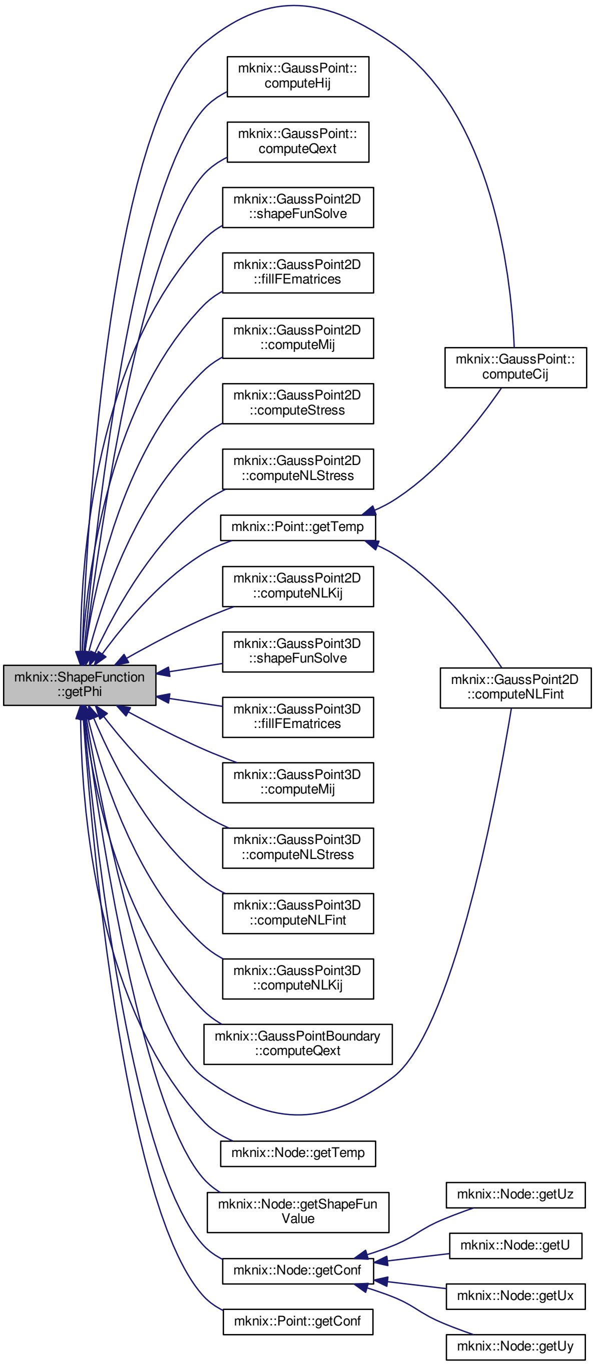
Here is the caller graph for this function:


mknix::ShapeFunction ::calc
mknix::GaussPoint:: shapeFunSolve
mknix::GaussPoint2D ::shapeFunSolve
mknix::GaussPoint3D ::shapeFunSolve
mknix::GaussPoint2D ::fillFEmatrices
mknix::GaussPoint3D ::fillFEmatrices
mknix::GaussPointBoundary ::shapeFunSolve
mknix::Point::shapeFunSolve

6.55.3.2 virtual double mknix::ShapeFunction::getPhi (int *i,* int *j*) [inline], [virtual]

Definition at line 61 of file shapefunction.h.

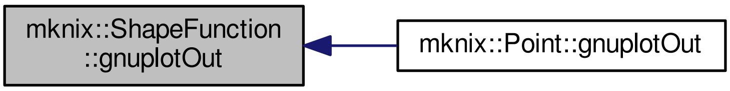
Here is the caller graph for this function:


mknix::ShapeFunction ::getPhi
mknix::GaussPoint:: computeHij
mknix::GaussPoint:: computeQext
mknix::GaussPoint2D ::shapeFunSolve
mknix::GaussPoint2D ::fillFEmatrices
mknix::GaussPoint2D ::computeMij
mknix::GaussPoint2D ::computeStress
mknix::GaussPoint2D ::computeNLStress
mknix::Point::getTemp
mknix::GaussPoint2D ::computeNLKij
mknix::GaussPoint3D ::shapeFunSolve
mknix::GaussPoint3D ::fillFEmatrices
mknix::GaussPoint3D ::computeMij
mknix::GaussPoint3D ::computeNLStress
mknix::GaussPoint3D ::computeNLFint
mknix::GaussPoint3D ::computeNLKij
mknix::GaussPointBoundary ::computeQext
mknix::Node::getTemp
mknix::Node::getShapeFun Value
mknix::Node::getConf
mknix::Point::getConf
mknix::GaussPoint:: computeCij
mknix::GaussPoint2D ::computeNLFint
mknix::Node::getUz
mknix::Node::getU
mknix::Node::getUx
mknix::Node::getUy

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 lmx::DenseMatrix<double> mknix::ShapeFunction::phi [protected]

Definition at line 47 of file shapefunction.h.

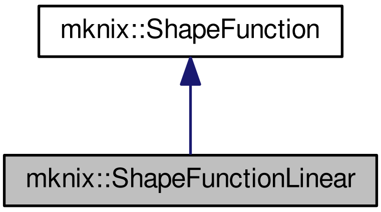
The documentation for this class was generated from the following files:

- shapefunction.h
- shapefunction.cpp

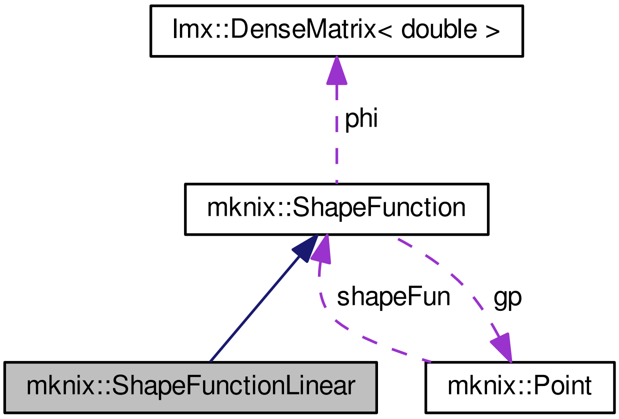
6.56 mknix::ShapeFunctionLinear Class Reference

```
#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.

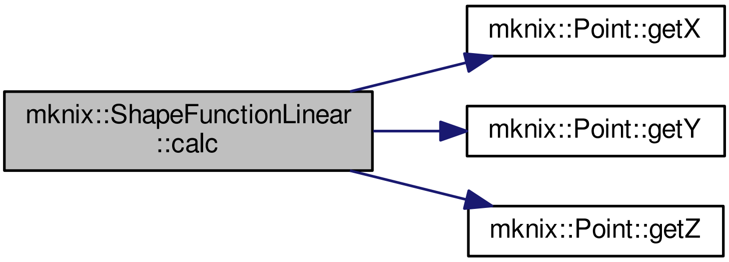
6.56.3 Member Function Documentation

6.56.3.1 void mknix::ShapeFunctionLinear::calc ( ) [virtual]

Implements mknix::ShapeFunction.

Definition at line 45 of file shapefunctionlinear.cpp.

Here is the call graph for this function:



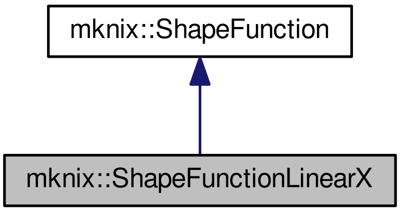
The documentation for this class was generated from the following files:

- shapefunctionlinear.h
- shapefunctionlinear.cpp

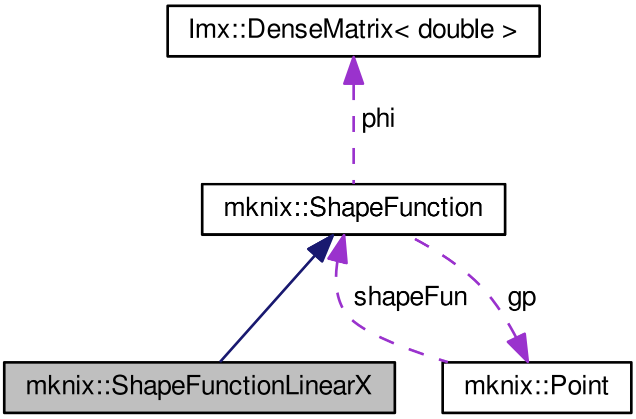
6.57 mknix::ShapeFunctionLinearX Class Reference

```
#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.

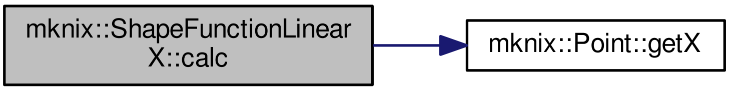
6.57.3 Member Function Documentation

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:



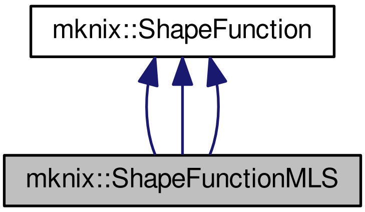
The documentation for this class was generated from the following files:

- shapefunctionlinear-x.h
- shapefunctionlinear-x.cpp

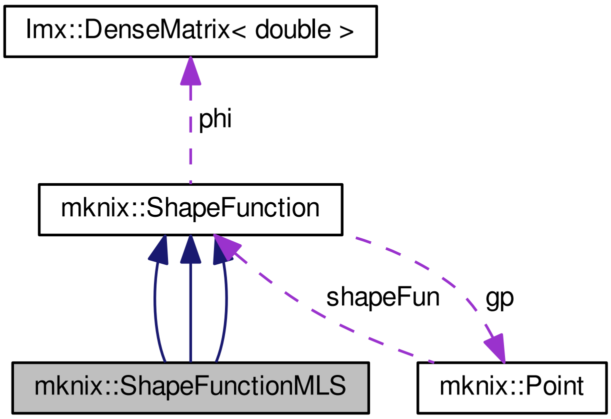
6.58 mknix::ShapeFunctionMLS Class Reference

```
#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 & 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 & , double & , 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 & , double & , 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:

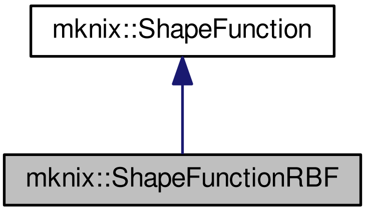
- shapefunctionMLS.h

- shapefunctionMLS2D.h
- shapefunctionMLS3D.h
- shapefunctionMLS.cpp
- shapefunctionMLS2D.cpp
- shapefunctionMLS3D.cpp

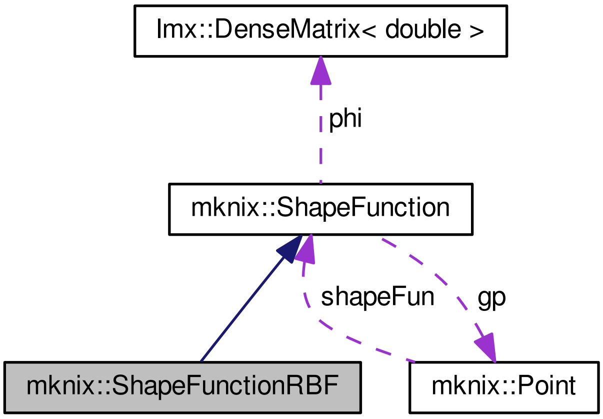
6.59 mknix::ShapeFunctionRBF Class Reference

```
#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 & 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.

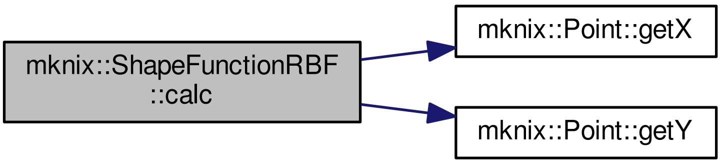
6.59.3 Member Function Documentation

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:



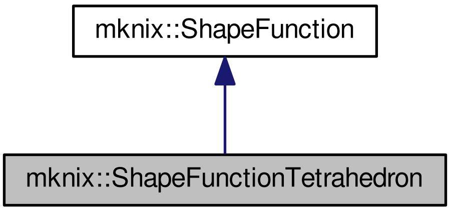
The documentation for this class was generated from the following files:

- shapefunctionRBF.h
- shapefunctionRBF.cpp

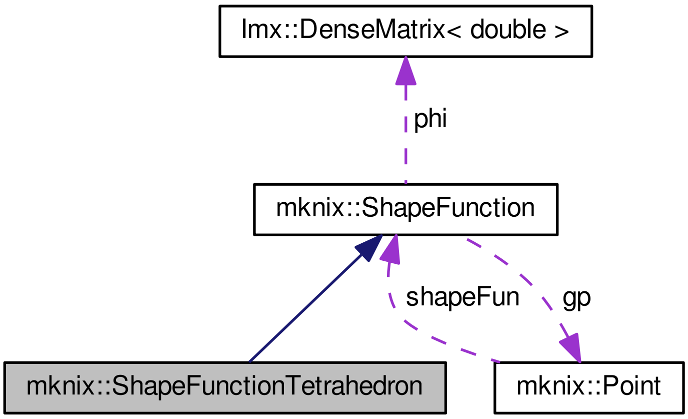
6.60 mknix::ShapeFunctionTetrahedron Class Reference

```
#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 & 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.

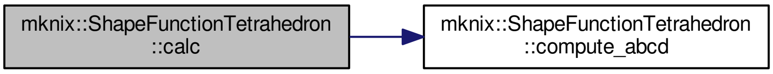
6.60.3 Member Function Documentation

6.60.3.1 void mknix::ShapeFunctionTetrahedron::calc ( ) [virtual]

Implements mknix::ShapeFunction.

Definition at line 46 of file shapefunctiontetrahedron.cpp.

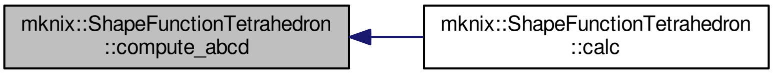
Here is the call graph for this function:


mknix::ShapeFunctionTetrahedron ::calc
mknix::ShapeFunctionTetrahedron ::compute_abcd

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:


mknix::ShapeFunctionTetrahedron ::compute_abcd
mknix::ShapeFunctionTetrahedron ::calc

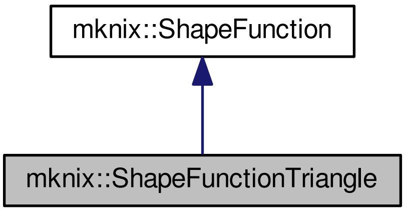
The documentation for this class was generated from the following files:

- shapefunctiontetrahedron.h
- shapefunctiontetrahedron.cpp

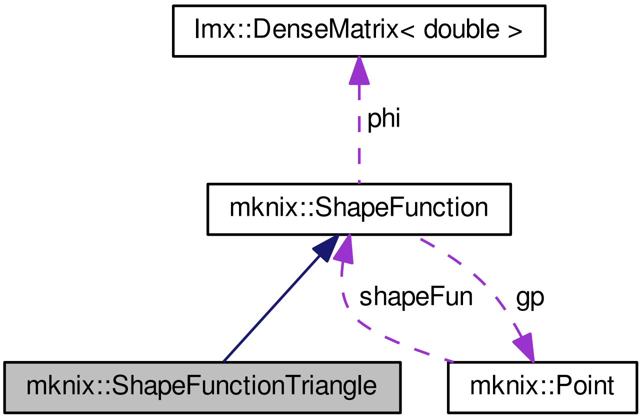
6.61 mknix::ShapeFunctionTriangle Class Reference

```
#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.

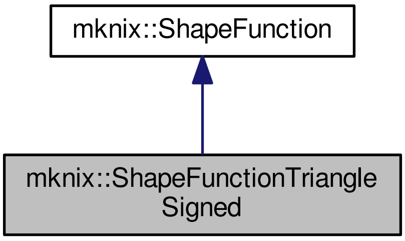
The documentation for this class was generated from the following files:

- shapefunctiontriangle.h
- shapefunctiontriangle.cpp

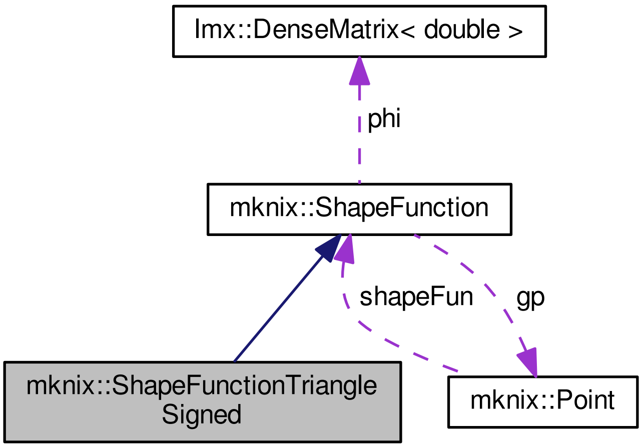
6.62 mknix::ShapeFunctionTriangleSigned Class Reference

`#include <shapefunctiontriangle3D.h>`

Inheritance diagram for mknix::ShapeFunctionTriangleSigned:



Collaboration diagram for mknix::ShapeFunctionTriangleSigned:


lmx::DenseMatrix< double >
phi
mknix::ShapeFunction
shapeFun
gp
mknix::ShapeFunctionTriangle Signed
mknix::Point

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 mknix::ShapeFunctionTriangleSigned::ShapeFunctionTriangleSigned ( )

Definition at line 26 of file shapefunctiontriangle3D.cpp.

6.62.2.2 mknix::ShapeFunctionTriangleSigned::ShapeFunctionTriangleSigned (Point ∗ *gp_in*)

Definition at line 33 of file shapefunctiontriangle3D.cpp.

6.62.2.3 mknix::ShapeFunctionTriangleSigned::∼ShapeFunctionTriangleSigned ( )

Definition at line 40 of file shapefunctiontriangle3D.cpp.

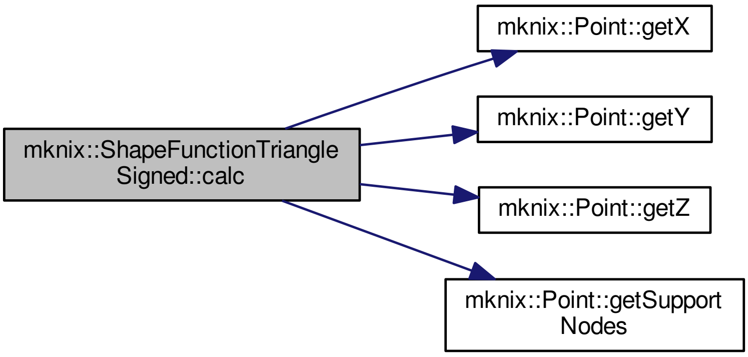
6.62.3 Member Function Documentation

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:


mknix::ShapeFunctionTriangle Signed::calc
mknix::Point::getX
mknix::Point::getY
mknix::Point::getZ
mknix::Point::getSupport Nodes

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 (lmx::Vector< data_type > &residue, lmx::Vector< data_type > &q)
- void staticThermalTangent (lmx::Matrix< data_type > &tangent_in, lmx::Vector< data_type > &q)
- bool staticThermalConvergence (lmx::Vector< data_type > &res, lmx::Vector< data_type > &q)
- void explicitThermalEvaluation (const lmx::Vector< data_type > &qt, lmx::Vector< data_type > &qtdot, double time)
- void dynamicThermalEvaluation (const lmx::Vector< data_type > &q, lmx::Vector< data_type > &qdot, double time)
- void dynamicThermalResidue (lmx::Vector< data_type > &residue, const lmx::Vector< data_type > &q, const lmx::Vector< data_type > &qdot, double time)
- void dynamicThermalTangent (lmx::Matrix< data_type > &tangent_in, const lmx::Vector< data_type > &q, double partial_qdot, double time)
- bool dynamicThermalConvergence (const lmx::Vector< data_type > &q, const lmx::Vector< data_type > &qdot, double time)
- bool dynamicThermalConvergenceInThermomechanical (const lmx::Vector< data_type > &q, const lmx::↩ Vector< data_type > &qdot, double time)
- void explicitAcceleration (const lmx::Vector< data_type > &q, const lmx::Vector< data_type > &qdot, lmx↩ ::Vector< data_type > &qddot, double time)
- void dynamicAcceleration (const lmx::Vector< data_type > &q, const lmx::Vector< data_type > &qdot, lmx↩ ::Vector< data_type > &qddot, double time)
- void dynamicResidue (lmx::Vector< data_type > &residue, const lmx::Vector< data_type > &q, const lmx↩ ::Vector< data_type > &qdot, const lmx::Vector< data_type > &qddot, double time)
- void dynamicTangent (lmx::Matrix< data_type > &tangent_in, const lmx::Vector< data_type > &q, const lmx::Vector< data_type > &qdot, double partial_qdot, double partial_qddot, double time)
- bool dynamicConvergence (const lmx::Vector< data_type > &q, const lmx::Vector< data_type > &qdot, const lmx::Vector< data_type > &qddot, double time)
- void staticResidue (lmx::Vector< data_type > &residue, lmx::Vector< data_type > &q)
- void staticTangent (lmx::Matrix< data_type > &tangent_in, lmx::Vector< data_type > &q)
- bool staticConvergence (lmx::Vector< data_type > &res, lmx::Vector< data_type > &q)
- void stepTriggered ()
- void writeConfStep ()
- lmx::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 & 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.

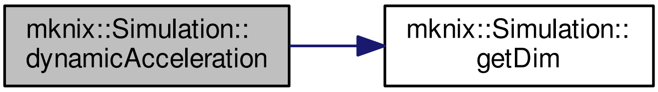
6.63.2.3 mknix::Simulation::Simulation (const Simulation &) [delete]

6.63.3 Member Function Documentation

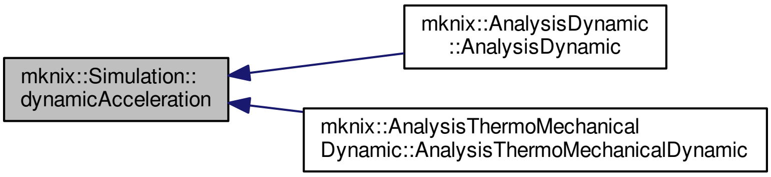
6.63.3.1 void mknix::Simulation::dynamicAcceleration (const lmx::Vector< data_type > & *q*, const lmx::Vector< data_type > & *qdot*, lmx::Vector< data_type > & *qddot*, double *time*)

Definition at line 736 of file simulation.cpp.

Here is the call graph for this function:



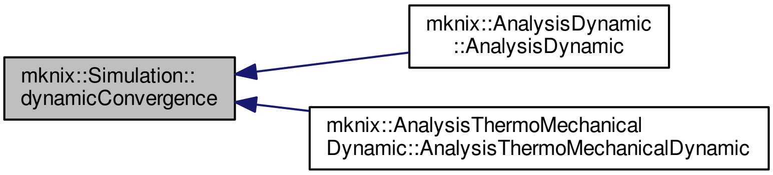
Here is the caller graph for this function:



6.63.3.2 bool mknix::Simulation::dynamicConvergence (const lmx::Vector< data_type > & *q*, const lmx::Vector< data_type > & *qdot*, const lmx::Vector< data_type > & *qddot*, double *time*)

Definition at line 814 of file simulation.cpp.

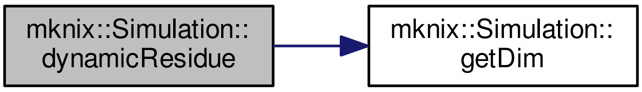
Here is the caller graph for this function:



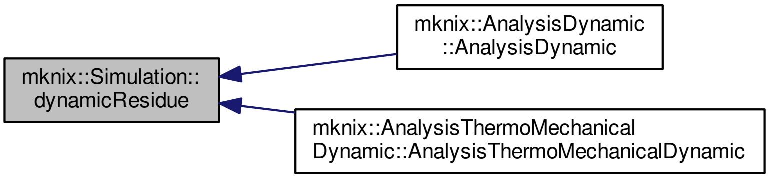
6.63.3.3 void mknix::Simulation::dynamicResidue (lmx::Vector< data_type > & *residue*, const lmx::Vector< data_type > & *q*, const lmx::Vector< data_type > & *qdot*, const lmx::Vector< data_type > & *qddot*, double *time*)

Definition at line 764 of file simulation.cpp.

Here is the call graph for this function:



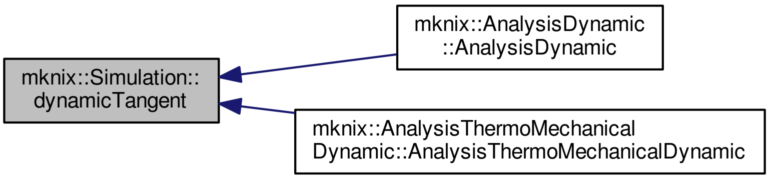
Here is the caller graph for this function:



6.63.3.4 void mknix::Simulation::dynamicTangent (lmx::Matrix< data_type > & *tangent_in*, const lmx::Vector< data_type > & *q*, const lmx::Vector< data_type > & *qdot*, double *partial_qdot*, double *partial_qddot*, double *time*)

Definition at line 800 of file simulation.cpp.

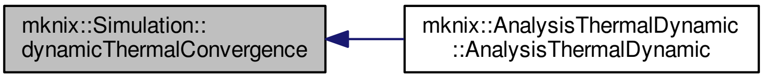
Here is the caller graph for this function:



6.63.3.5 bool mknix::Simulation::dynamicThermalConvergence (const lmx::Vector< data_type > & *q,* const lmx::Vector< data_type > & *qdot,* double *time*)

Definition at line 649 of file simulation.cpp.

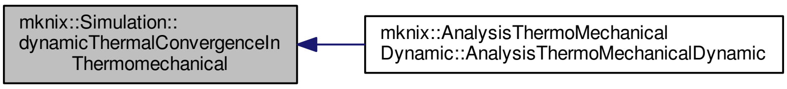
Here is the caller graph for this function:



6.63.3.6 bool mknix::Simulation::dynamicThermalConvergenceInThermomechanical (const lmx::Vector< data_type > & *q,* const lmx::Vector< data_type > & *qdot,* double *time*)

Definition at line 685 of file simulation.cpp.

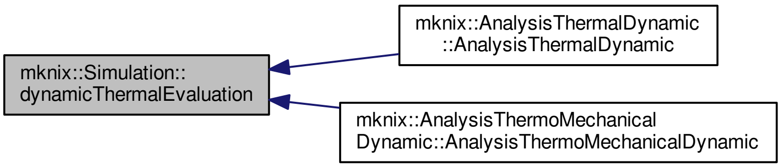
Here is the caller graph for this function:


mknix::Simulation::
dynamicThermalConvergenceIn
Thermomechanical
mknix::AnalysisThermoMechanical
Dynamic::AnalysisThermoMechanicalDynamic

6.63.3.7 void mknix::Simulation::dynamicThermalEvaluation (const lmx::Vector< data_type > & *q,* lmx::Vector< data_type > & *qdot,* double *time*)

Definition at line 561 of file simulation.cpp.

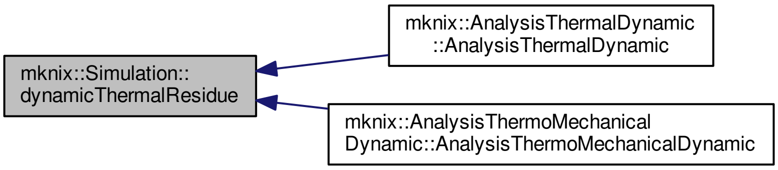
Here is the caller graph for this function:


mknix::Simulation::
dynamicThermalEvaluation
mknix::AnalysisThermalDynamic
::AnalysisThermalDynamic
mknix::AnalysisThermoMechanical
Dynamic::AnalysisThermoMechanicalDynamic

6.63.3.8 void mknix::Simulation::dynamicThermalResidue (lmx::Vector< data_type > & *residue,* const lmx::Vector< data_type > & *q,* const lmx::Vector< data_type > & *qdot,* double *time*)

Definition at line 593 of file simulation.cpp.

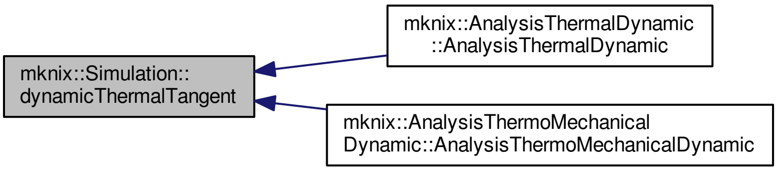
Here is the caller graph for this function:


mknix::Simulation::
dynamicThermalResidue
mknix::AnalysisThermalDynamic
::AnalysisThermalDynamic
mknix::AnalysisThermoMechanical
Dynamic::AnalysisThermoMechanicalDynamic

6.63.3.9 void mknix::Simulation::dynamicThermalTangent (lmx::Matrix< data_type > & *tangent_in,* const lmx::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:


mknix::Simulation::
dynamicThermalTangent
mknix::AnalysisThermalDynamic
::AnalysisThermalDynamic
mknix::AnalysisThermoMechanical
Dynamic::AnalysisThermoMechanicalDynamic

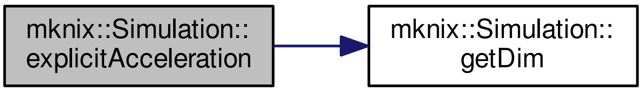
6.63.3.10 void mknix::Simulation::endSimulation ( )

Definition at line 227 of file simulation.cpp.

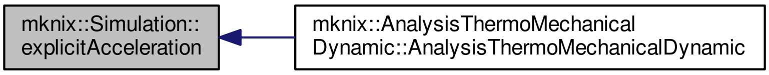
6.63.3.11 void mknix::Simulation::explicitAcceleration (const lmx::Vector< data_type > & *q*, const lmx::Vector< data_type > & *qdot*, lmx::Vector< data_type > & *qddot*, double *time*)

Definition at line 704 of file simulation.cpp.

Here is the call graph for this function:



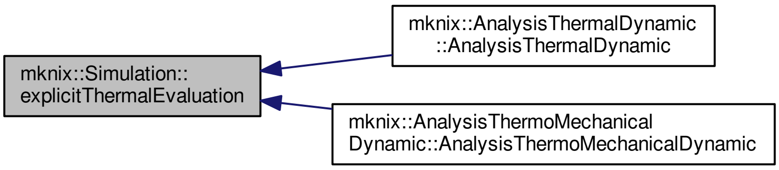
Here is the caller graph for this function:


mknix::Simulation::
explicitAcceleration
mknix::AnalysisThermoMechanical
Dynamic::AnalysisThermoMechanicalDynamic

6.63.3.12 void mknix::Simulation::explicitThermalEvaluation (const lmx::Vector< data_type > & *qt*, lmx::Vector< data_type > & *qtdot*, double *time*)

Definition at line 531 of file simulation.cpp.

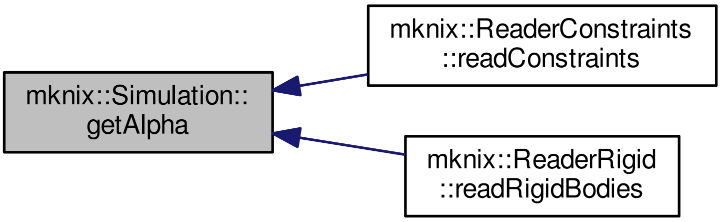
Here is the caller graph for this function:


mknix::AnalysisThermalDynamic
::AnalysisThermalDynamic
mknix::Simulation::
explicitThermalEvaluation
mknix::AnalysisThermoMechanical
Dynamic::AnalysisThermoMechanicalDynamic

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:



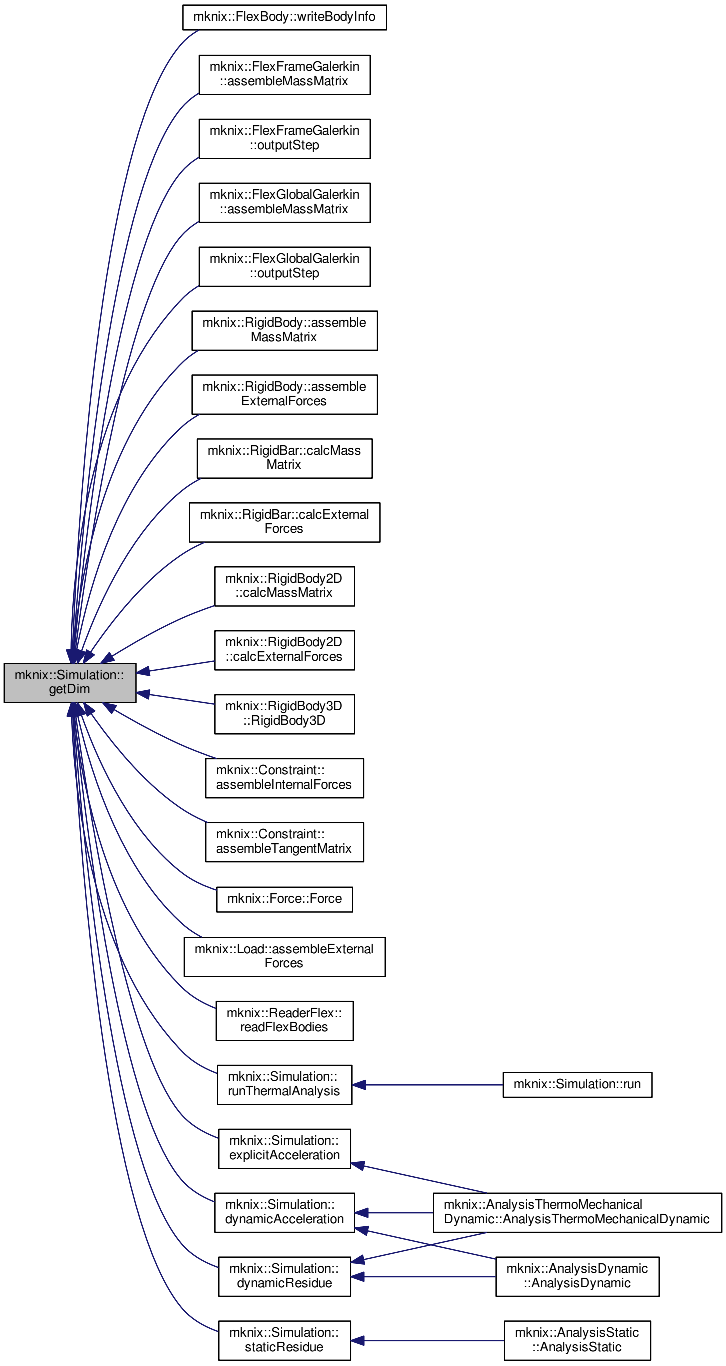
6.63.3.14 static std::string mknix::Simulation::getConstraintMethod ( ) [inline], [static]

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.

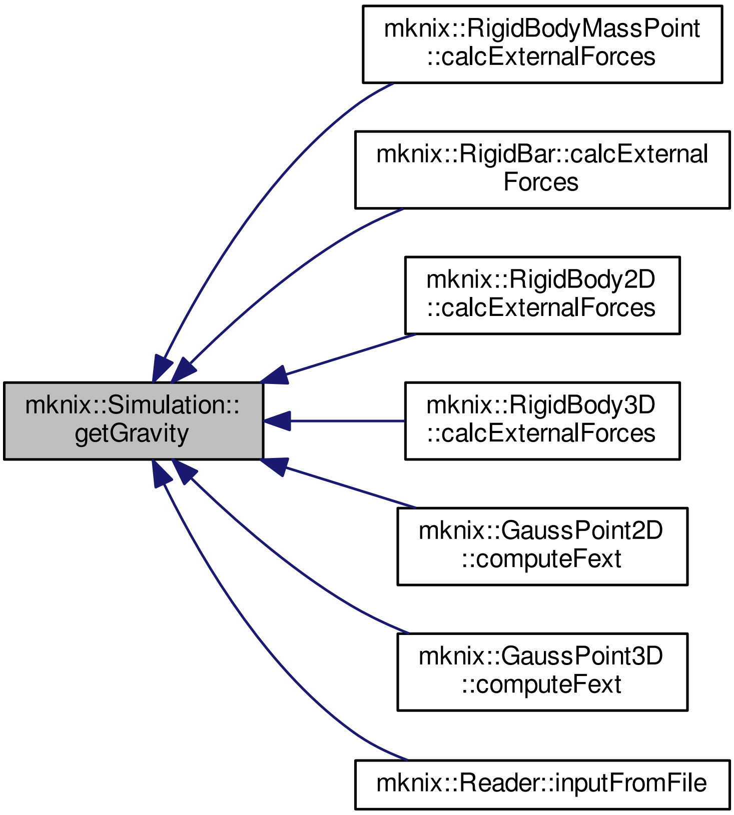
Here is the caller graph for this function:


mknix::FlexBody::writeBodyInfo
mknix::FlexFrameGalerkin ::assembleMassMatrix
mknix::FlexFrameGalerkin ::outputStep
mknix::FlexGlobalGalerkin ::assembleMassMatrix
mknix::FlexGlobalGalerkin ::outputStep
mknix::RigidBody::assemble MassMatrix
mknix::RigidBody::assemble ExternalForces
mknix::RigidBar::calcMass Matrix
mknix::RigidBar::calcExternal Forces
mknix::RigidBody2D ::calcMassMatrix
mknix::RigidBody2D ::calcExternalForces
mknix::Simulation:: getDim
mknix::RigidBody3D ::RigidBody3D
mknix::Constraint:: assembleInternalForces
mknix::Constraint:: assembleTangentMatrix
mknix::Force::Force
mknix::Load::assembleExternal Forces
mknix::ReaderFlex:: readFlexBodies
mknix::Simulation:: runThermalAnalysis
mknix::Simulation::run
mknix::Simulation:: explicitAcceleration
mknix::Simulation:: dynamicAcceleration
mknix::AnalysisThermoMechanical Dynamic::AnalysisThermoMechanicalDynamic
mknix::Simulation:: dynamicResidue
mknix::AnalysisDynamic ::AnalysisDynamic
mknix::Simulation:: staticResidue
mknix::AnalysisStatic ::AnalysisStatic

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:


mknix::RigidBodyMassPoint ::calcExternalForces
mknix::RigidBar::calcExternal Forces
mknix::RigidBody2D ::calcExternalForces
mknix::Simulation:: getGravity
mknix::RigidBody3D ::calcExternalForces
mknix::GaussPoint2D ::computeFext
mknix::GaussPoint3D ::computeFext
mknix::Reader::inputFromFile

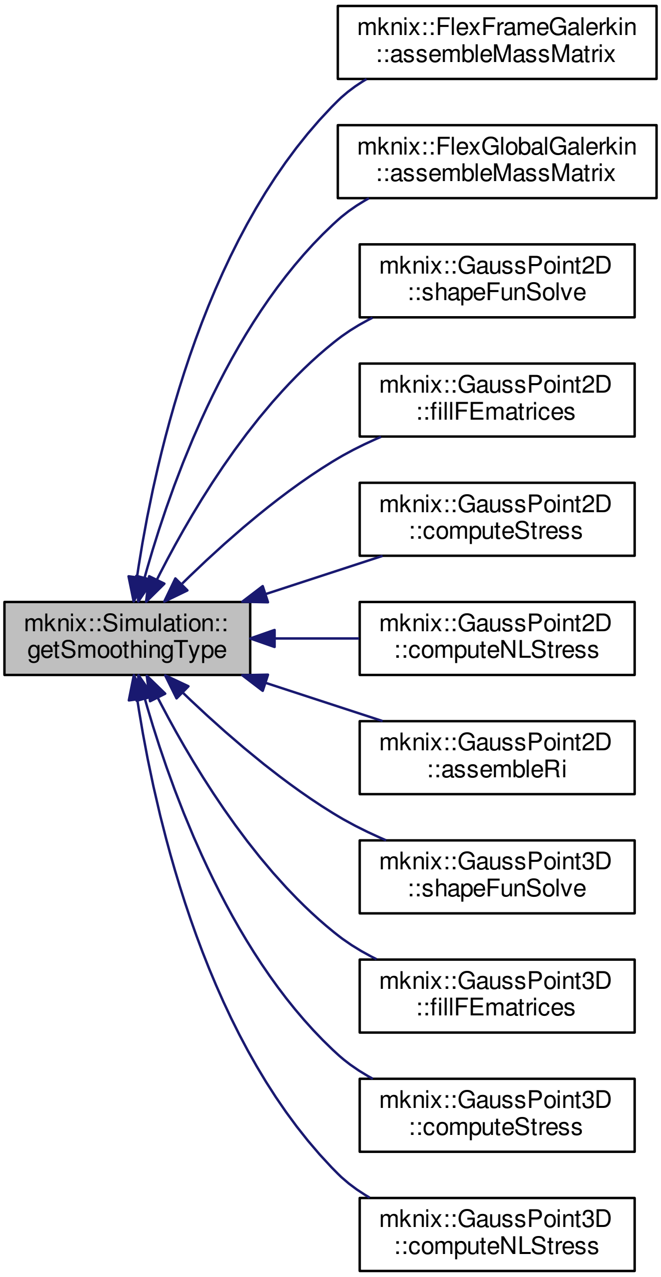
6.63.3.17 std::vector< double > mknix::Simulation::getInterfaceNodesCoords ( )

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.

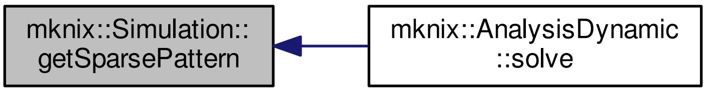
Here is the caller graph for this function:



6.63.3.19 lmx::DenseMatrix<data_type>& mknix::Simulation::getSparsePattern ( ) [inline]

Definition at line 190 of file simulation.h.

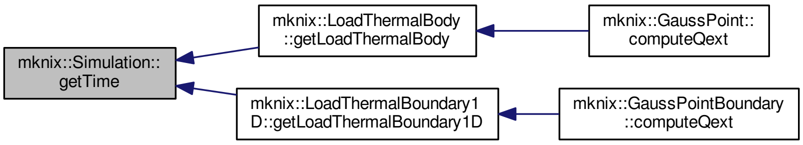
Here is the caller graph for this function:



6.63.3.20 static double mknix::Simulation::getTime ( ) [inline], [static]

Definition at line 205 of file simulation.h.

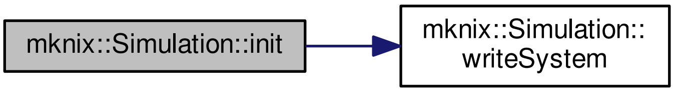
Here is the caller graph for this function:



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 & *fileIn*)

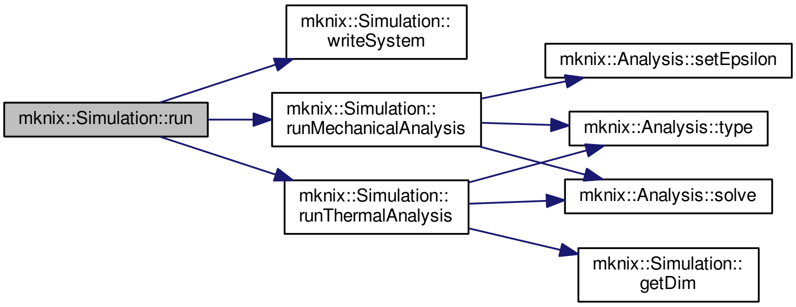
Definition at line 72 of file simulation.cpp.

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.

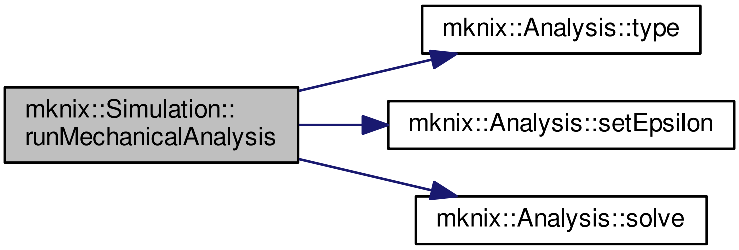
Here is the call graph for this function:


mknix::Simulation::run
mknix::Simulation::writeSystem
mknix::Simulation::runMechanicalAnalysis
mknix::Simulation::runThermalAnalysis
mknix::Analysis::setEpsilon
mknix::Analysis::type
mknix::Analysis::solve
mknix::Simulation::getDim

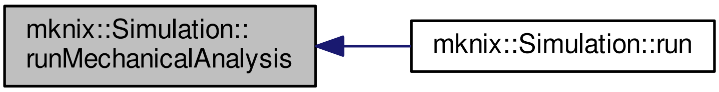
6.63.3.25 void mknix::Simulation::runMechanicalAnalysis (**Analysis** ∗ *theAnalysis_in*)

Definition at line 351 of file simulation.cpp.

Here is the call graph for this function:


mknix::Simulation::runMechanicalAnalysis
mknix::Analysis::type
mknix::Analysis::setEpsilon
mknix::Analysis::solve

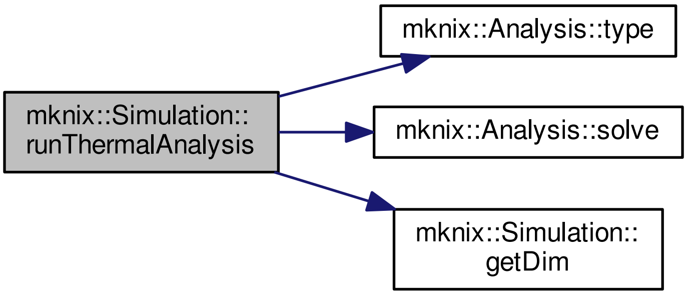
Here is the caller graph for this function:



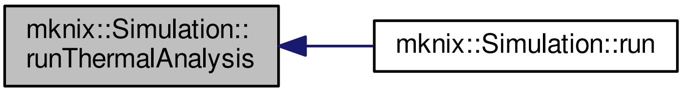
6.63.3.26 void mknix::Simulation::runThermalAnalysis (**Analysis** ∗ *theAnalysis_in*)

Definition at line 303 of file simulation.cpp.

Here is the call graph for this function:



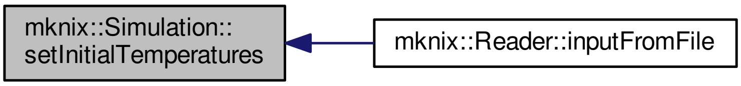
Here is the caller graph for this function:



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:



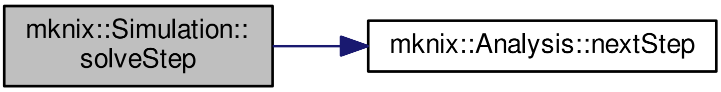
6.63.3.28 void mknix::Simulation::setOutputFilesDetail (int *level_in*) [inline]

Definition at line 76 of file simulation.h.

6.63.3.29 void mknix::Simulation::solveStep ( )

Definition at line 213 of file simulation.cpp.

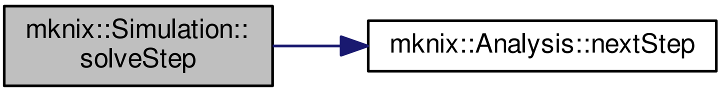
Here is the call graph for this function:



6.63.3.30 void mknix::Simulation::solveStep (double $*$ *signal*, double $*$ *o_output* = 0)

Definition at line 218 of file simulation.cpp.

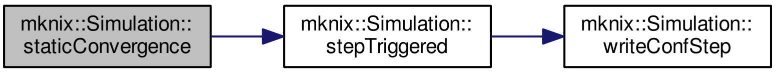
Here is the call graph for this function:



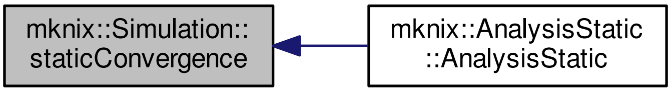
6.63.3.31 bool mknix::Simulation::staticConvergence (lmx::Vector$<$ data_type $>$ & *res*, lmx::Vector$<$ data_type $>$ & *q*)

Definition at line 892 of file simulation.cpp.

Here is the call graph for this function:



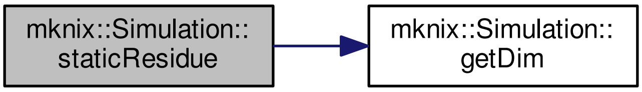
Here is the caller graph for this function:



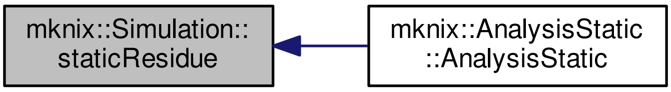
6.63.3.32 void mknix::Simulation::staticResidue (lmx::Vector< data_type > & *residue*, lmx::Vector< data_type > & *q*)

Definition at line 856 of file simulation.cpp.

Here is the call graph for this function:



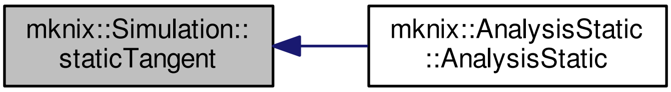
Here is the caller graph for this function:



6.63.3.33 void mknix::Simulation::staticTangent (lmx::Matrix< data_type > & *tangent_in*, lmx::Vector< data_type > & *q*)

Definition at line 882 of file simulation.cpp.

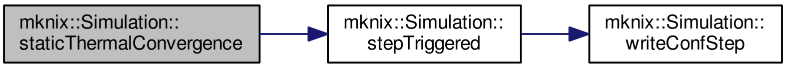
Here is the caller graph for this function:



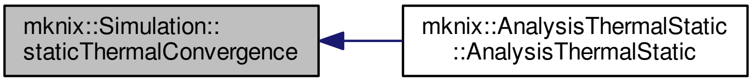
6.63.3.34 bool mknix::Simulation::staticThermalConvergence (lmx::Vector< data_type > & *res,* lmx::Vector< data_type > & *q*)

Definition at line 509 of file simulation.cpp.

Here is the call graph for this function:



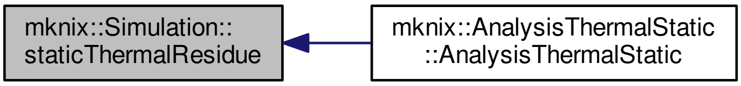
Here is the caller graph for this function:



6.63.3.35 void mknix::Simulation::staticThermalResidue (lmx::Vector< data_type > & *residue,* lmx::Vector< data_type > & *q*)

Definition at line 469 of file simulation.cpp.

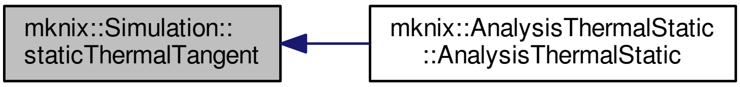
Here is the caller graph for this function:



6.63.3.36 void mknix::Simulation::staticThermalTangent (lmx::Matrix< data_type > & *tangent_in*, lmx::Vector< data_type > & *q*)

Definition at line 497 of file simulation.cpp.

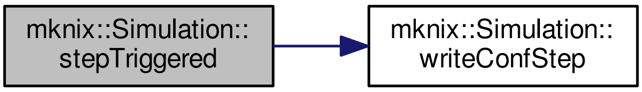
Here is the caller graph for this function:



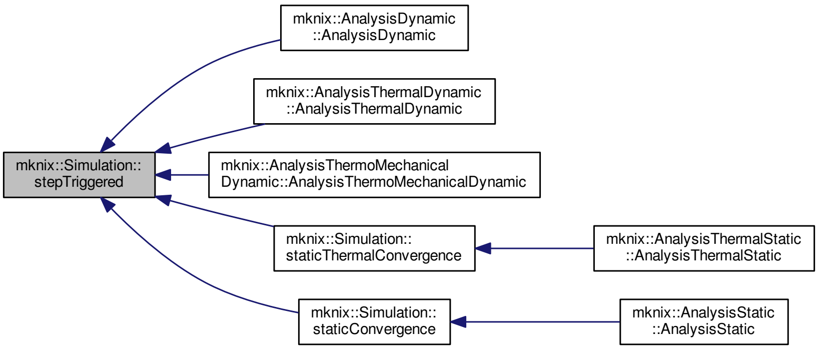
6.63.3.37 void mknix::Simulation::stepTriggered ( )

Definition at line 914 of file simulation.cpp.

Here is the call graph for this function:



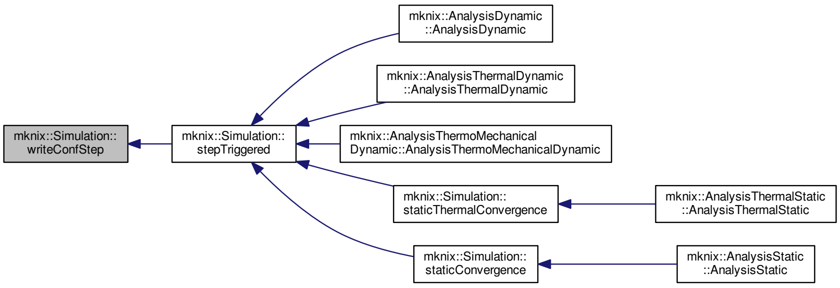
Here is the caller graph for this function:



6.63.3.38 void mknix::Simulation::writeConfStep ( )

Definition at line 940 of file simulation.cpp.

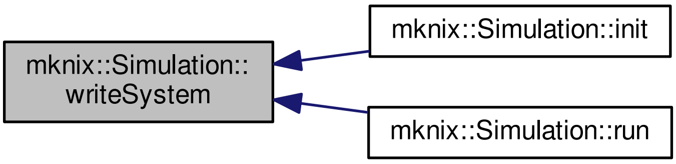
Here is the caller graph for this function:



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.

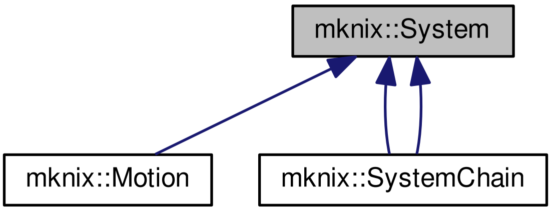
The documentation for this class was generated from the following files:

- 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 (lmx::Matrix< data_type > &)
- void assembleConductivityMatrix (lmx::Matrix< data_type > &)
- void assembleExternalHeat (lmx::Vector< data_type > &)
- void assembleInternalHeat (lmx::Vector< data_type > &)
- void assembleThermalTangentMatrix (lmx::Matrix< data_type > &)
- void calcMassMatrix ()

- void calcInternalForces ()
- void calcExternalForces ()
- void calcTangentMatrix ()
- void assembleMassMatrix (lmx::Matrix< data_type > &)
- void assembleInternalForces (lmx::Vector< data_type > &)
- void assembleExternalForces (lmx::Vector< data_type > &)
- void assembleTangentMatrix (lmx::Matrix< data_type > &)
- void assembleConstraintForces (lmx::Vector< data_type > &)
- void setMechanical ()
- void outputStep (const lmx::Vector< data_type > &, const lmx::Vector< data_type > &)
- void outputStep (const lmx::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 (lmx::Matrix< data_type > & *globalCapacity_in*)

Definition at line 232 of file system.cpp.

6.64.3.2 void mknix::System::assembleConductivityMatrix (lmx::Matrix< data_type > & *globalConductivity_in*)

Definition at line 244 of file system.cpp.

6.64.3.3 void mknix::System::assembleConstraintForces (lmx::Vector< data_type > & *internalForces_in*)

Definition at line 440 of file system.cpp.

6.64.3.4 void mknix::System::assembleExternalForces (lmx::Vector< data_type > & *externalForces_in*)

Definition at line 401 of file system.cpp.

6.64.3.5 void mknix::System::assembleExternalHeat (lmx::Vector< data_type > & *externalHeat_in*)

Definition at line 264 of file system.cpp.

6.64.3.6 void mknix::System::assembleInternalForces (lmx::Vector< data_type > & *internalForces_in*)

Definition at line 385 of file system.cpp.

6.64.3.7 void mknix::System::assembleInternalHeat (lmx::Vector< data_type > & *internalHeat_in*)

Definition at line 281 of file system.cpp.

6.64.3.8 void mknix::System::assembleMassMatrix (lmx::Matrix< data_type > & *globalMass_in*)

Definition at line 369 of file system.cpp.

6.64.3.9 void mknix::System::assembleTangentMatrix (lmx::Matrix< data_type > & *globalTangent_in*)

Definition at line 424 of file system.cpp.

6.64.3.10 void mknix::System::assembleThermalTangentMatrix (lmx::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.

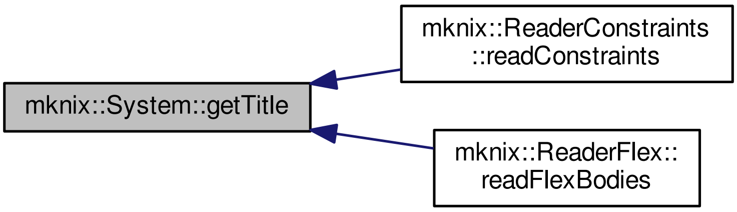
6.64.3.24 void mknix::System::getThermalNodes (std::vector< double > & *x_coordinates*)

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.

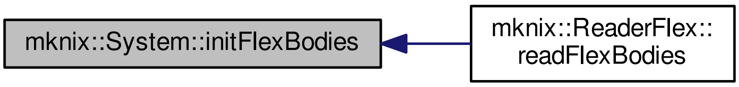
Here is the caller graph for this function:


mknix::ReaderConstraints ::readConstraints
mknix::System::getTitle
mknix::ReaderFlex:: readFlexBodies

6.64.3.26 void mknix::System::initFlexBodies ( )

Definition at line 123 of file system.cpp.

Here is the caller graph for this function:


mknix::System::initFlexBodies
mknix::ReaderFlex:: readFlexBodies

6.64.3.27 void mknix::System::outputStep (const lmx::Vector< data_type > & *q*, const lmx::Vector< data_type > & *qdot*)

Definition at line 477 of file system.cpp.

6.64.3.28 void mknix::System::outputStep (const lmx::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.

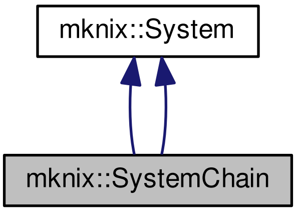
The documentation for this class was generated from the following files:

- system.h
- system.cpp

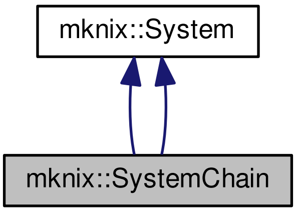
6.65 mknix::SystemChain Class Reference

`#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 > &timeLenghts_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 > &timeLenghts_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 > & *timeLenghts_in*) [inline]

Definition at line 75 of file systemchain2.h.

6.65.3.16 void mknix::SystemChain::setTimeLengths (std::map< double, double > & *timeLenghts_in*) [inline]

Definition at line 75 of file systemchain.h.

6.65.3.17 void mknix::SystemChain::update (double) [virtual]

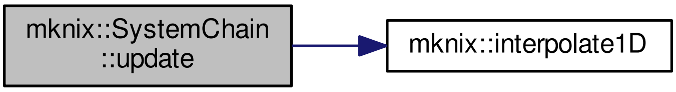
Reimplemented from mknix::System.

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:



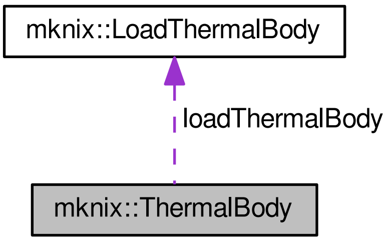
The documentation for this class was generated from the following files:

- 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 (lmx::Matrix< data_type > &)

  Assembles the local conductivity into the global matrix by calling each cell's cascade function.
- virtual void assembleConductivityMatrix (lmx::Matrix< data_type > &)

  Assembles the local conductivity into the global matrix by calling each cell's cascade function.
- virtual void assembleExternalHeat (lmx::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 lmx::Vector< data_type > &, const lmx::Vector< data_type > &)

  Postprocess and store step results for dynamic analysis.
- virtual void outputStep (const lmx::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< lmx::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 & 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

title_in	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 (lmx::Matrix< data_type > & *globalCapacity*) [virtual]

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

Parameters

globalCapacity	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 (lmx::Matrix< data_type > & *globalConductivity*) [virtual]

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

Parameters

global↵ Conductivity	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 (lmx::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 lmx::Vector< data_type > & *q*, const lmx::Vector< data_type > & *qdot*) [virtual]

Postprocess and store step results for dynamic analysis.

Parameters

q	Global configuration vector
qdot	Global configuration first derivative vector

Returns

void

Definition at line 250 of file bodythermal.cpp.

6.66.3.13 void mknix::ThermalBody::outputStep (const lmx::Vector< data_type > & *q*) [virtual]

Postprocess and store step results for static analysis.

Parameters

q	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

outFile	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

outputType_in	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< lmx::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 lmx::Vector< T > Class Template Reference

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

6.67.1 Detailed Description

template<typename T>class lmx::Vector< T >

Definition at line 39 of file cell.h.

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

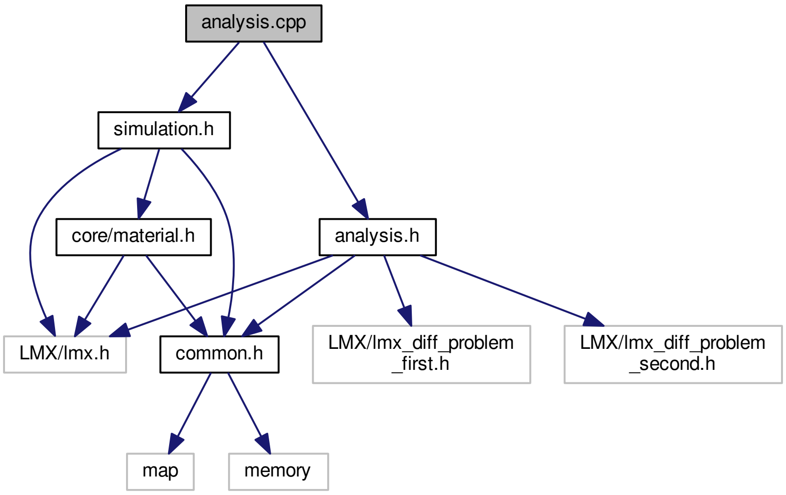
- cell.h

7 File Documentation

7.1 analysis.cpp File Reference

```
#include "analysis.h"
#include "simulation.h"
```

Include dependency graph for analysis.cpp:



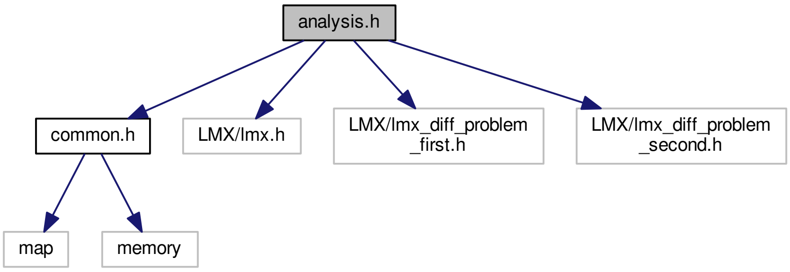
Namespaces

- mknix

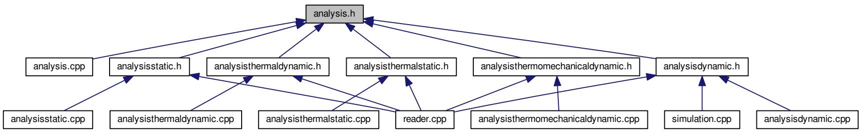
7.2 analysis.h File Reference

```
#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 mknix::Analysis

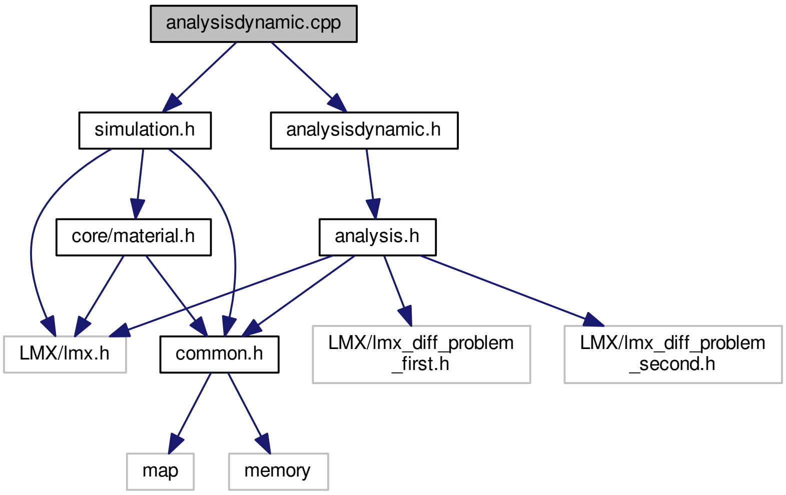
Namespaces

- mknix

7.3 analysisdynamic.cpp File Reference

```
#include "analysisdynamic.h"
#include "simulation.h"
```

Include dependency graph for analysisdynamic.cpp:



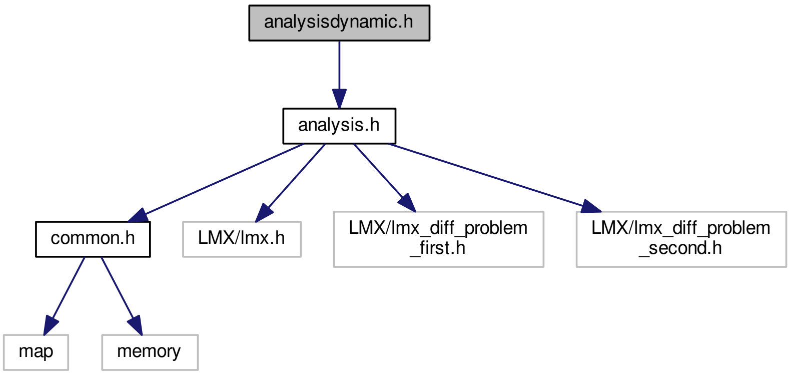
Namespaces

- mknix

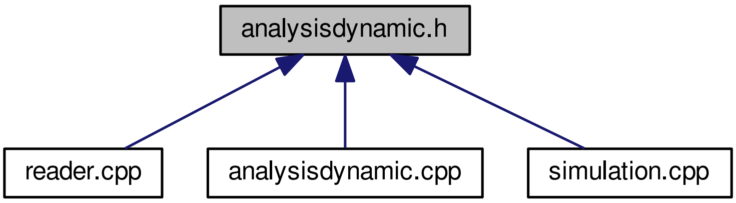
7.4 analysisdynamic.h File Reference

```
#include "analysis.h"
```

Include dependency graph for analysisdynamic.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::AnalysisDynamic

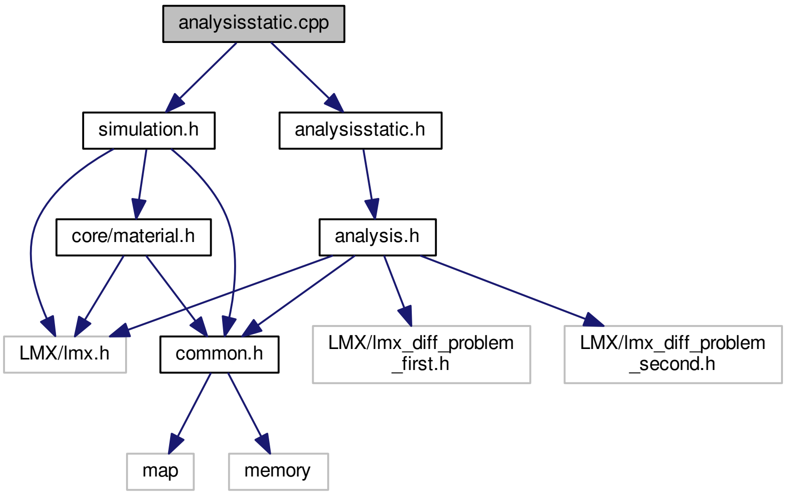
Namespaces

- mknix

7.5 analysisstatic.cpp File Reference

```
#include "analysisstatic.h"
#include "simulation.h"
```

Include dependency graph for analysisstatic.cpp:



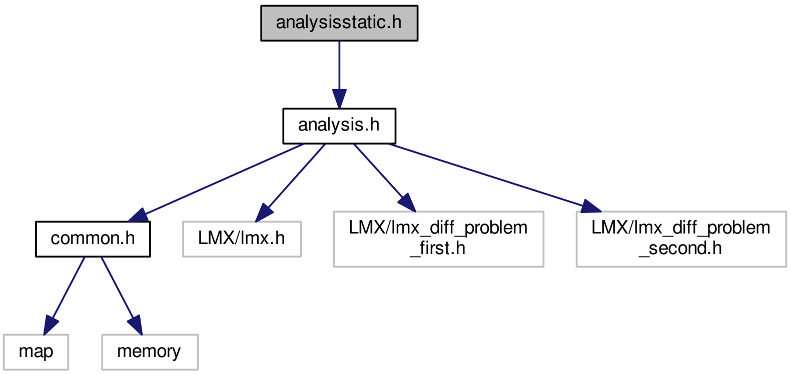
Namespaces

- mknix

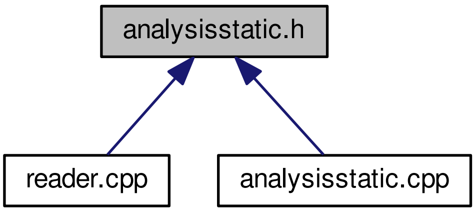
7.6 analysisstatic.h File Reference

```
#include "analysis.h"
```

Include dependency graph for analysisstatic.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::AnalysisStatic

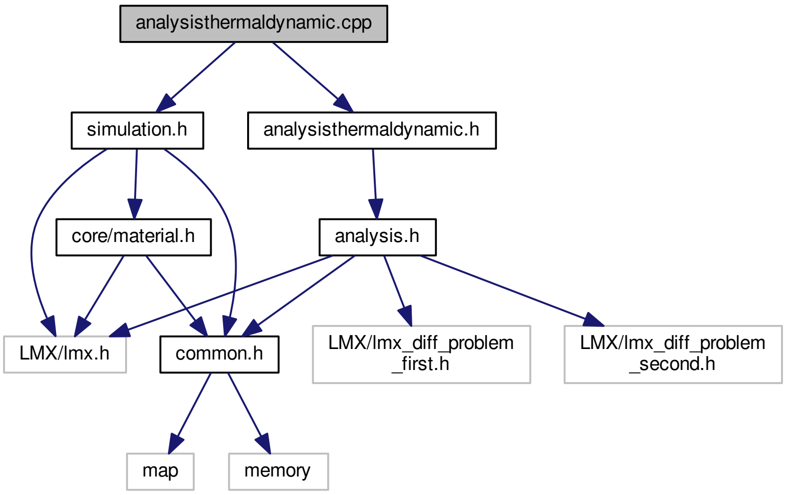
Namespaces

- mknix

7.7 analysisthermaldynamic.cpp File Reference

```
#include "analysisthermaldynamic.h"
#include "simulation.h"
```

Include dependency graph for analysisthermaldynamic.cpp:



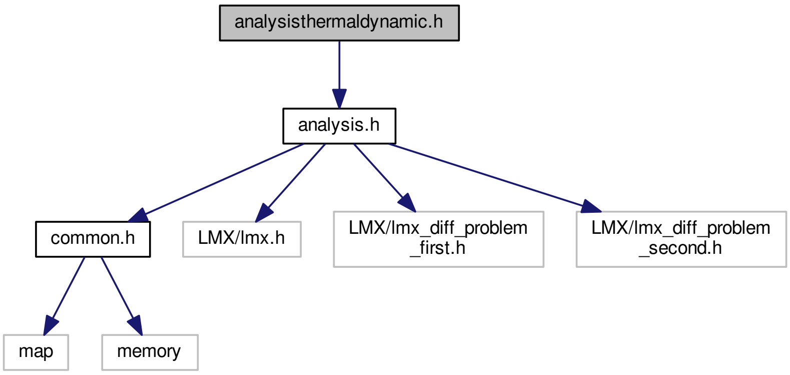
Namespaces

- mknix

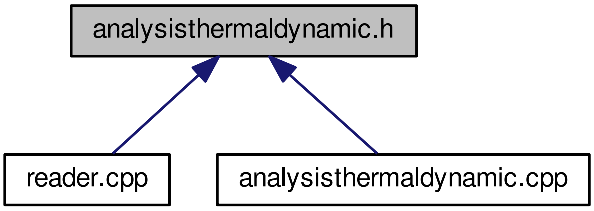
7.8 analysisthermaldynamic.h File Reference

```
#include "analysis.h"
```

Include dependency graph for analysisthermaldynamic.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::AnalysisThermalDynamic

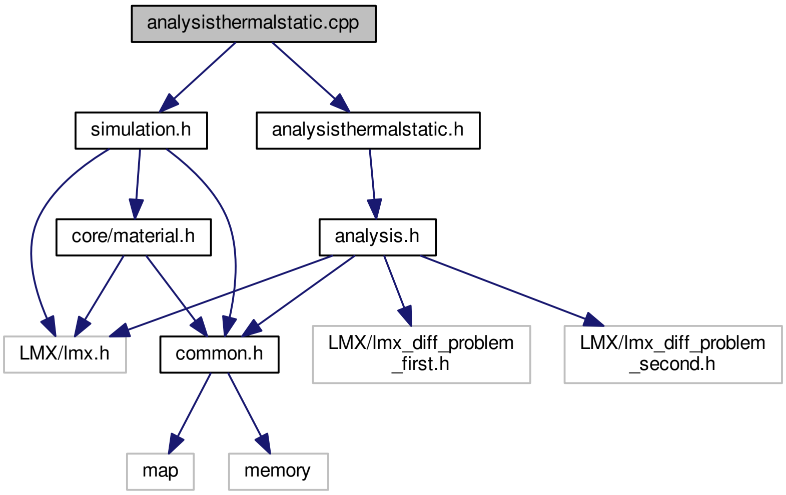
Namespaces

- mknix

7.9 analysisthermalstatic.cpp File Reference

```
#include "analysisthermalstatic.h"
#include "simulation.h"
```

Include dependency graph for analysisthermalstatic.cpp:



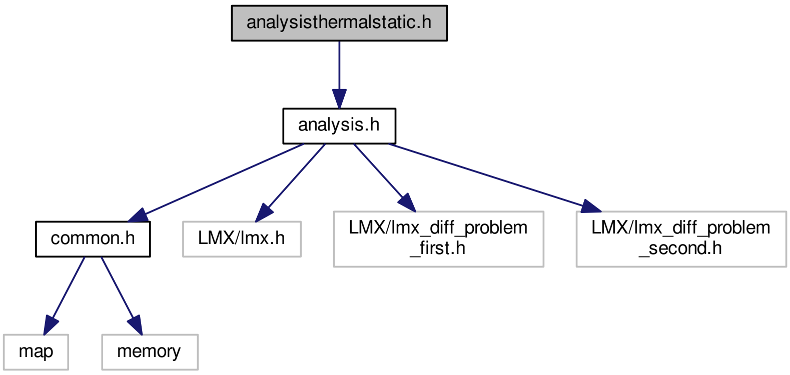
Namespaces

- mknix

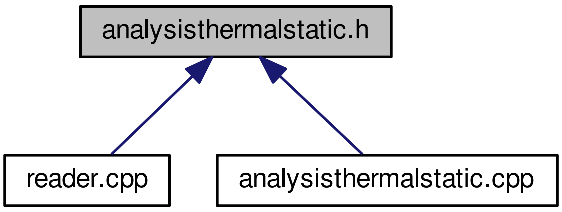
7.10 analysisthermalstatic.h File Reference

```
#include "analysis.h"
```

Include dependency graph for analysisthermalstatic.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::AnalysisThermalStatic

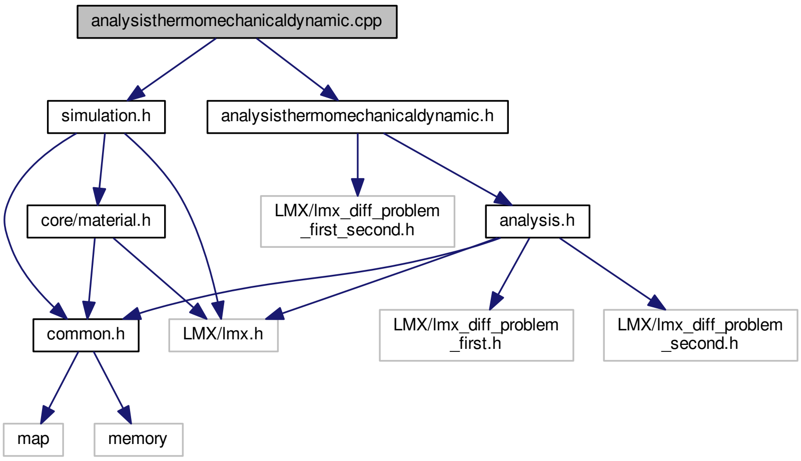
Namespaces

- mknix

7.11 analysisthermomechanicaldynamic.cpp File Reference

```
#include "analysisthermomechanicaldynamic.h"
#include "simulation.h"
```

Include dependency graph for analysisthermomechanicaldynamic.cpp:


analysisthermomechanicaldynamic.cpp
simulation.h
analysisthermomechanicaldynamic.h
core/material.h
LMX/lmx_diff_problem _first_second.h
analysis.h
common.h
LMX/lmx.h
LMX/lmx_diff_problem _first.h
LMX/lmx_diff_problem _second.h
map
memory

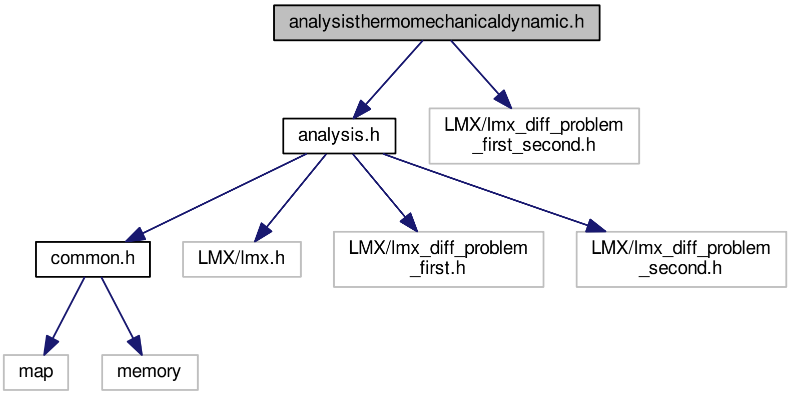
Namespaces

- mknix

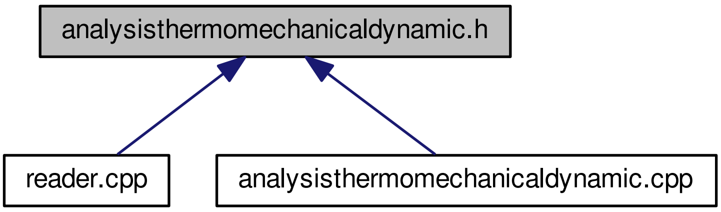
7.12 analysisthermomechanicaldynamic.h File Reference

```
#include "analysis.h"
#include <LMX/lmx_diff_problem_first_second.h>
```

Include dependency graph for analysisthermomechanicaldynamic.h:


analysisthermomechanicaldynamic.h
analysis.h
LMX/lmx_diff_problem _first_second.h
common.h
LMX/lmx.h
LMX/lmx_diff_problem _first.h
LMX/lmx_diff_problem _second.h
map
memory

This graph shows which files directly or indirectly include this file:


analysisthermomechanicaldynamic.h
reader.cpp
analysisthermomechanicaldynamic.cpp

Classes

- class mknix::AnalysisThermoMechanicalDynamic

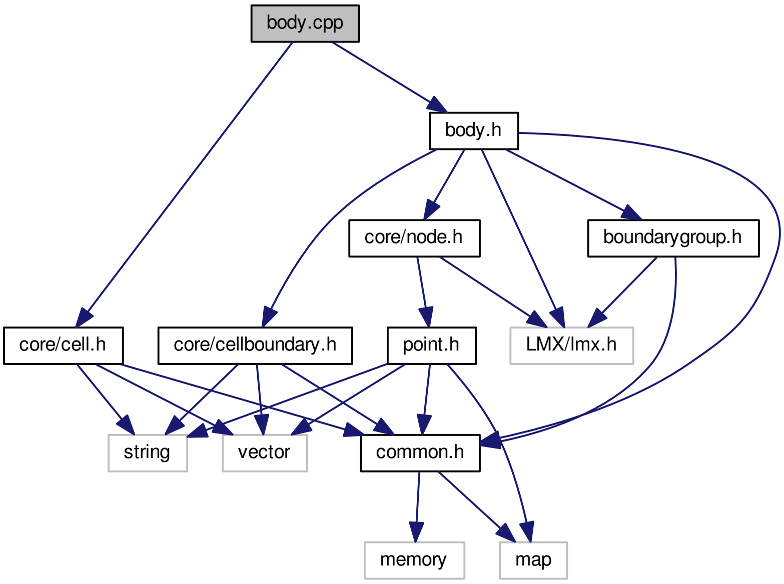
Namespaces

- mknix

7.13 body.cpp File Reference

```
#include "body.h"
#include <core/cell.h>
```

Include dependency graph for body.cpp:



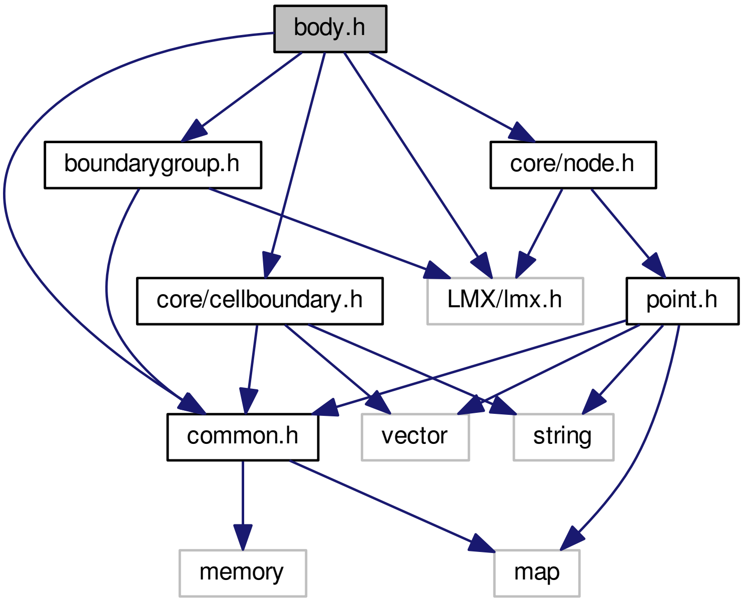
Namespaces

- mknix

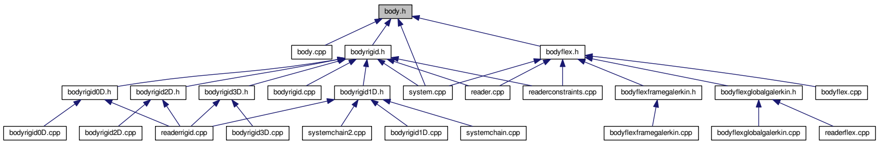
7.14 body.h File Reference

```
#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

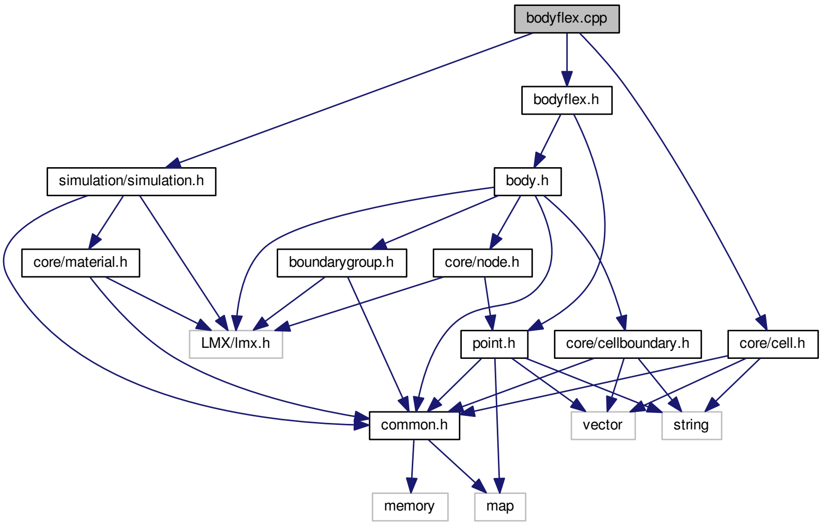
Namespaces

- mknix

7.15 bodyflex.cpp File Reference

```
#include "bodyflex.h"
#include <core/cell.h>
#include <simulation/simulation.h>
```

Include dependency graph for bodyflex.cpp:



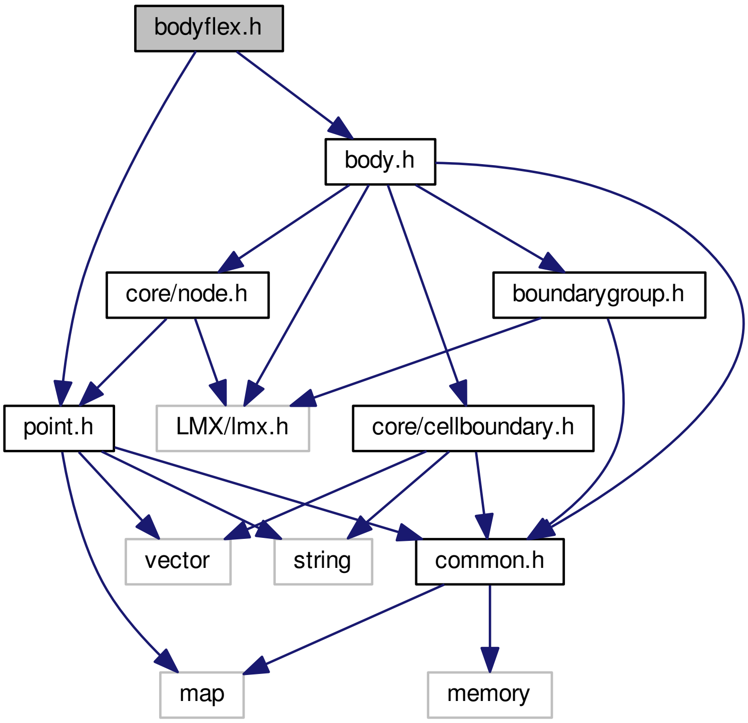
Namespaces

- mknix

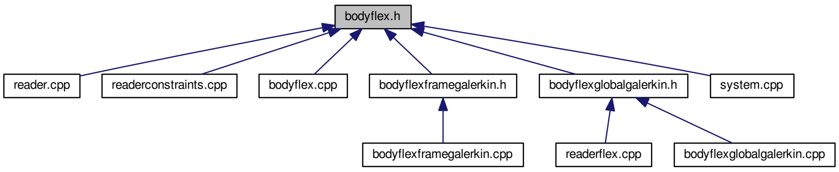
7.16 bodyflex.h File Reference

```
#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 mknix::FlexBody

Namespaces

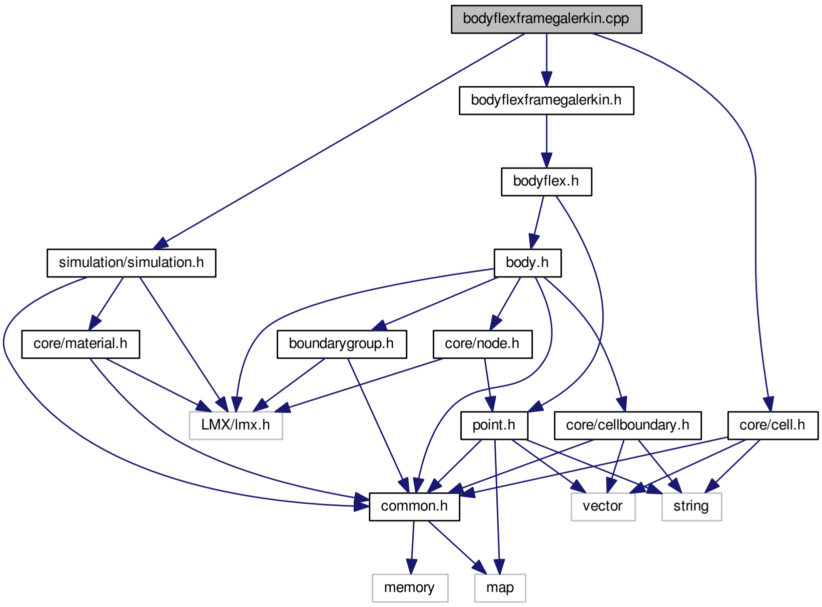
- mknix

7.17 bodyflexframegalerkin.cpp File Reference

```
#include "bodyflexframegalerkin.h"
```

```
#include <core/cell.h>
#include <simulation/simulation.h>
```

Include dependency graph for bodyflexframegalerkin.cpp:



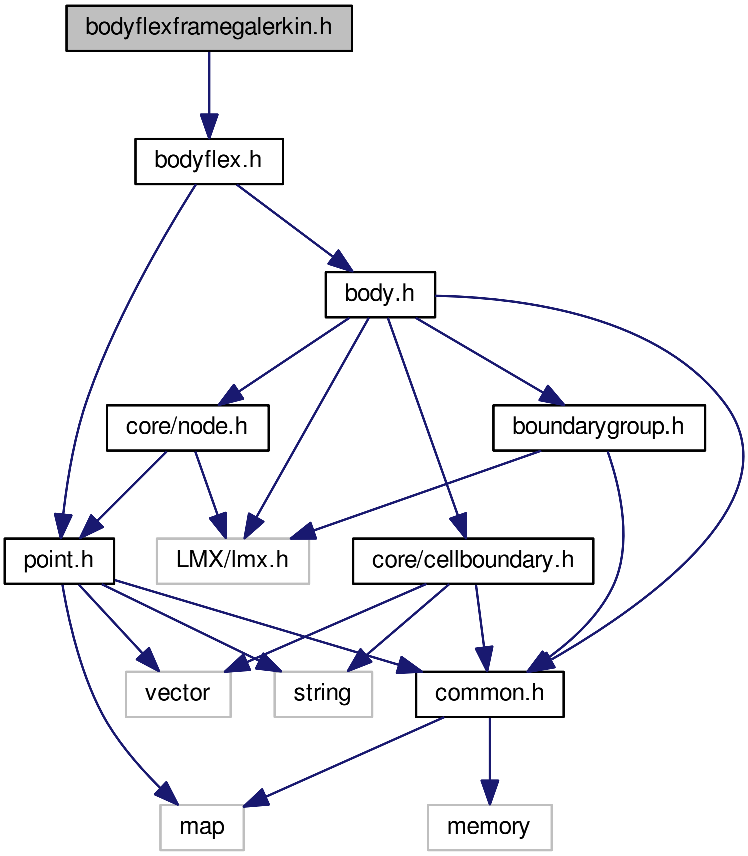
Namespaces

- mknix

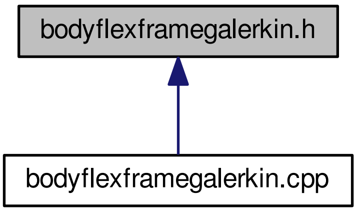
7.18 bodyflexframegalerkin.h File Reference

```
#include "bodyflex.h"
```

Include dependency graph for bodyflexframegalerkin.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::FlexFrameGalerkin

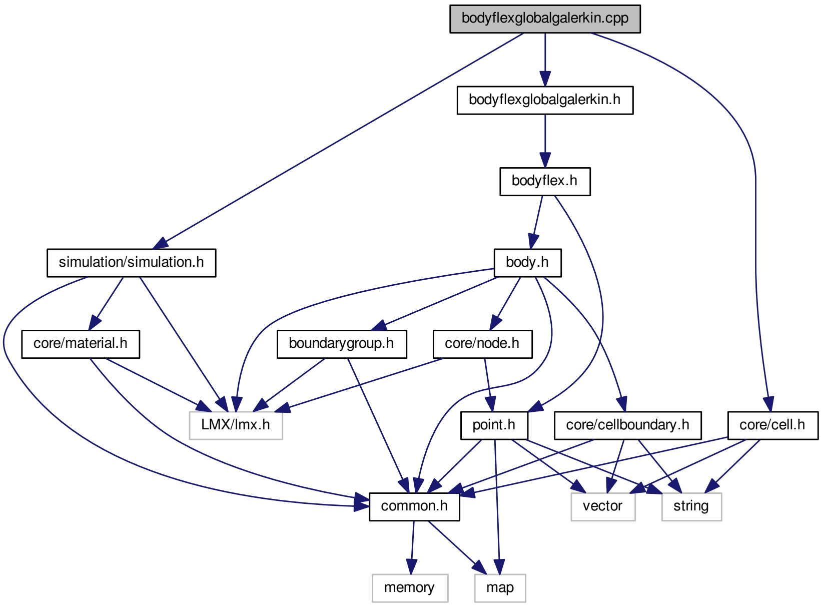
Namespaces

- mknix

7.19 bodyflexglobalgalerkin.cpp File Reference

```
#include "bodyflexglobalgalerkin.h"
#include <core/cell.h>
#include <simulation/simulation.h>
```

Include dependency graph for bodyflexglobalgalerkin.cpp:



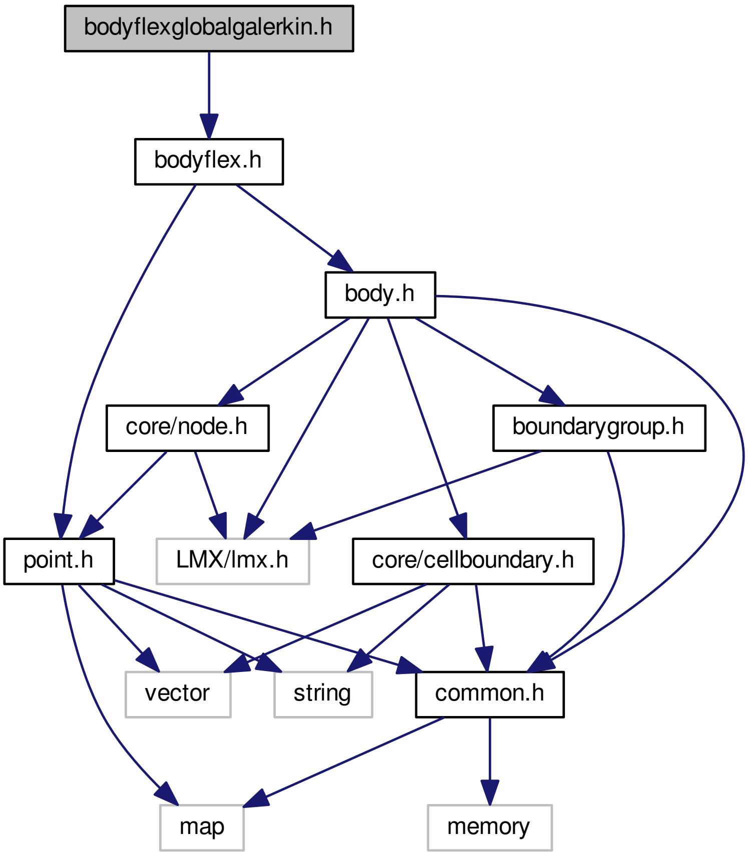
Namespaces

- mknix

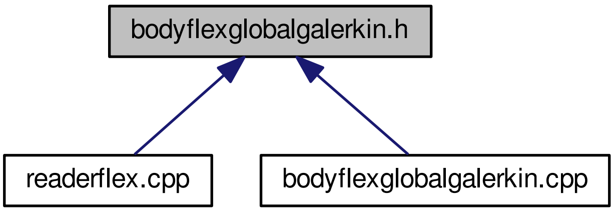
7.20 bodyflexglobalgalerkin.h File Reference

```
#include "bodyflex.h"
```

Include dependency graph for bodyflexglobalgalerkin.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::FlexGlobalGalerkin

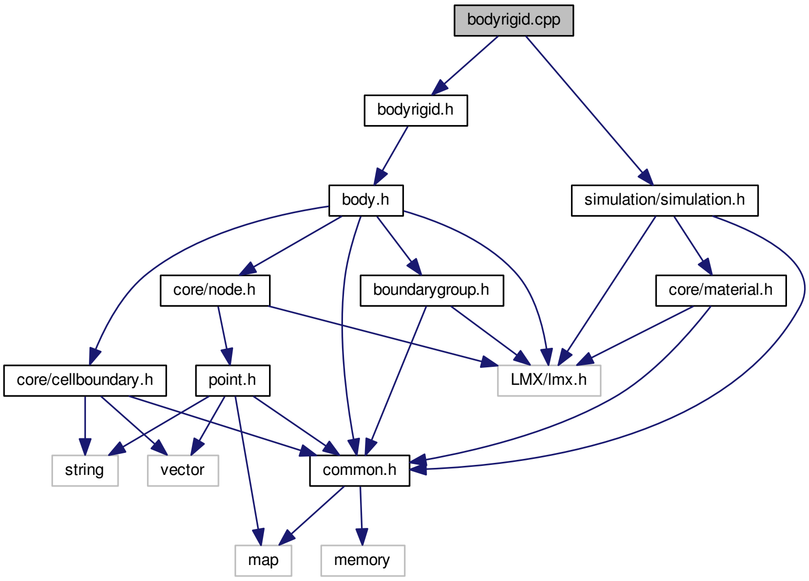
Namespaces

- mknix

7.21 bodyrigid.cpp File Reference

```
#include "bodyrigid.h"
#include <simulation/simulation.h>
```

Include dependency graph for bodyrigid.cpp:



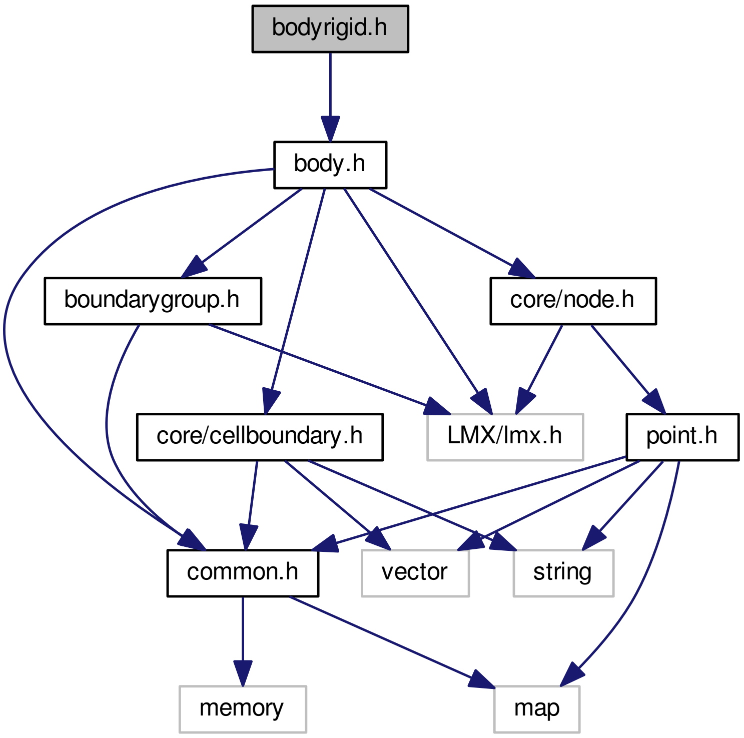
Namespaces

- mknix

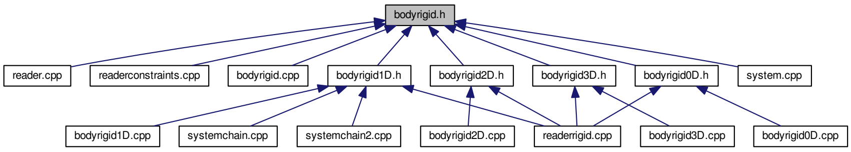
7.22 bodyrigid.h File Reference

```
#include "body.h"
```

Include dependency graph for bodyrigid.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::RigidBody

Namespaces

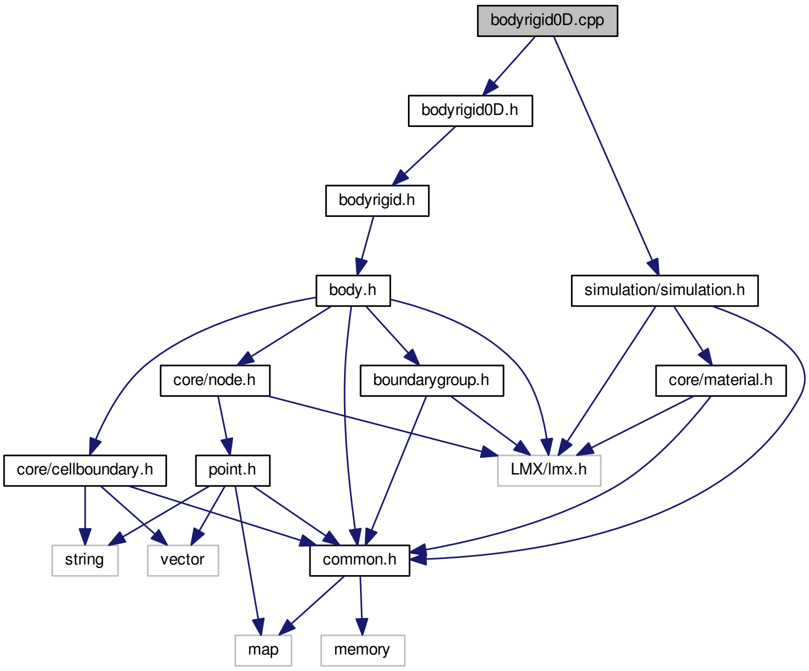
- mknix

7.23 bodyrigid0D.cpp File Reference

```
#include "bodyrigid0D.h"
```

```
#include <simulation/simulation.h>
```

Include dependency graph for bodyrigid0D.cpp:



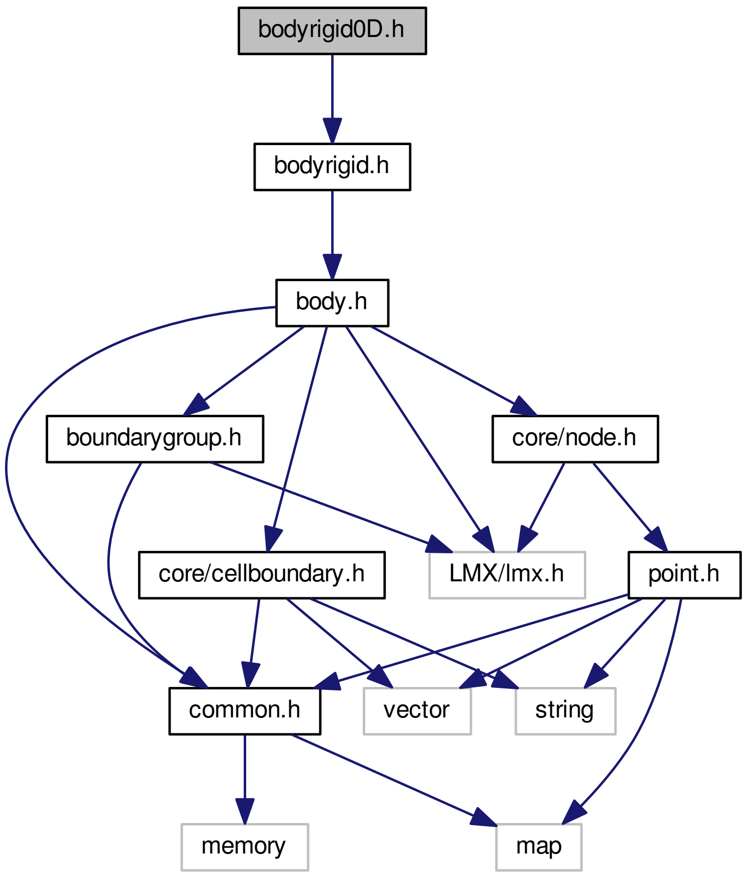
Namespaces

- mknix

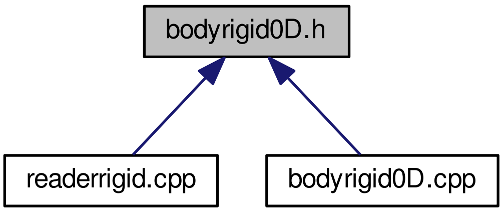
7.24 bodyrigid0D.h File Reference

```
#include "bodyrigid.h"
```

Include dependency graph for bodyrigid0D.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::RigidBodyMassPoint

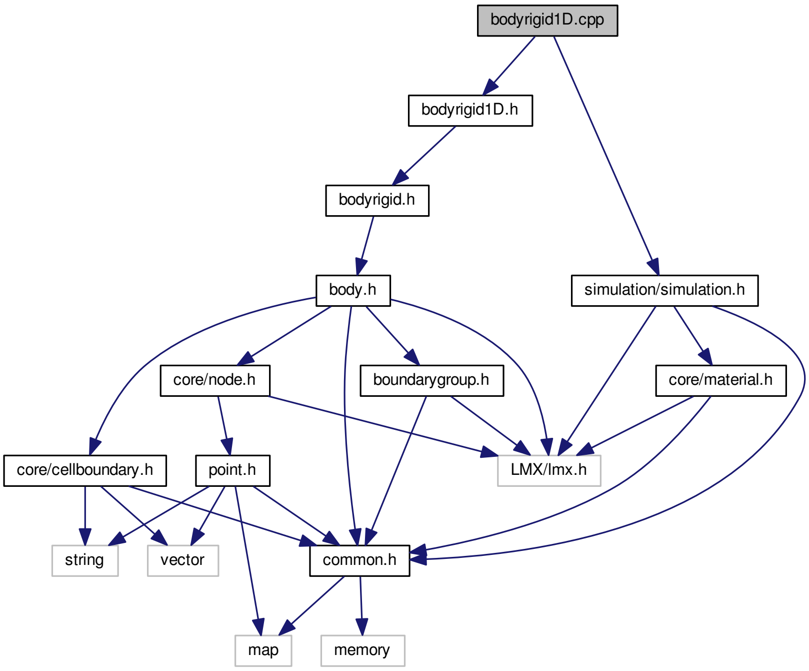
Namespaces

- mknix

7.25 bodyrigid1D.cpp File Reference

```
#include "bodyrigid1D.h"
#include <simulation/simulation.h>
```

Include dependency graph for bodyrigid1D.cpp:



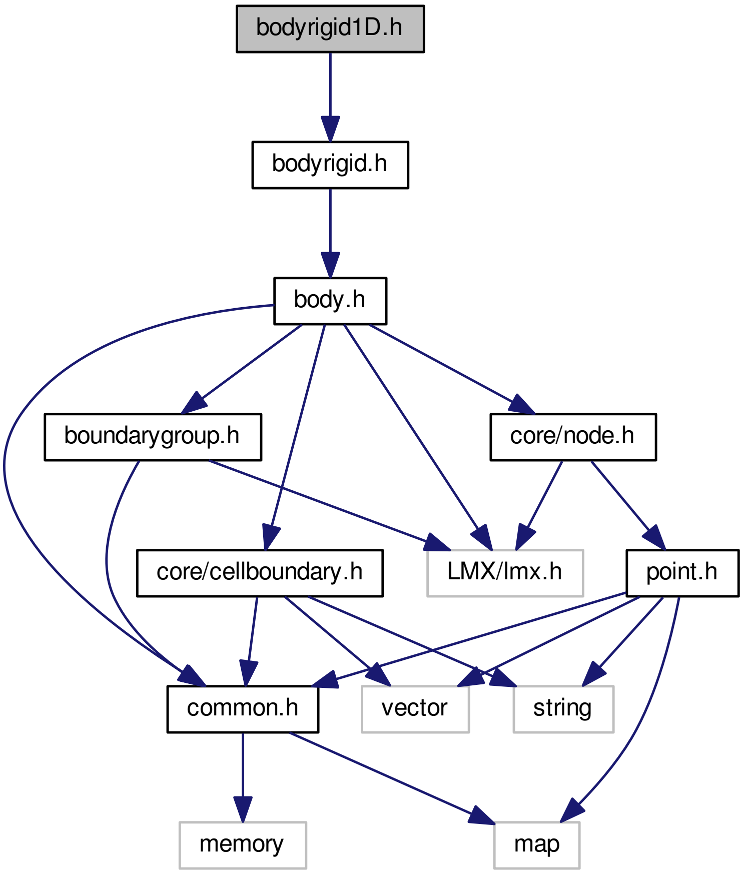
Namespaces

- mknix

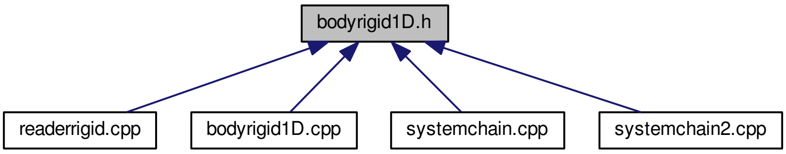
7.26 bodyrigid1D.h File Reference

```
#include "bodyrigid.h"
```

Include dependency graph for bodyrigid1D.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::RigidBar

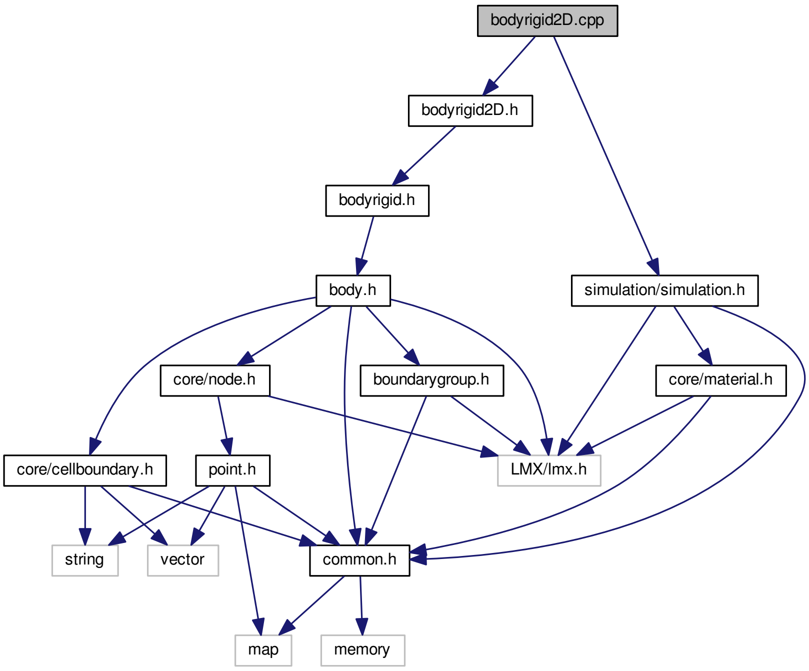
Namespaces

- mknix

7.27 bodyrigid2D.cpp File Reference

```
#include "bodyrigid2D.h"
#include <simulation/simulation.h>
```

Include dependency graph for bodyrigid2D.cpp:



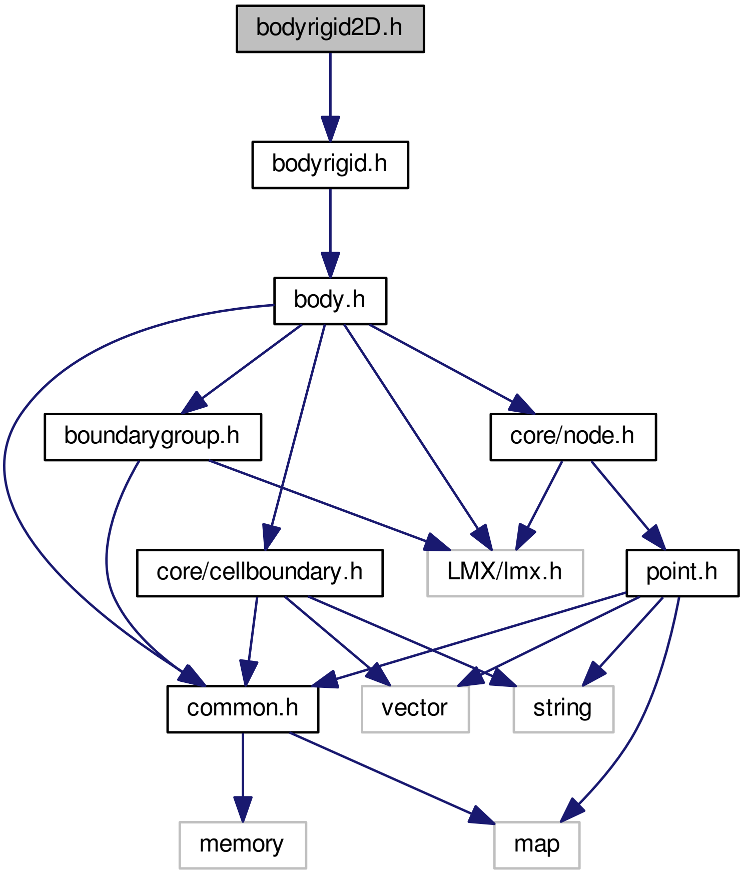
Namespaces

- mknix

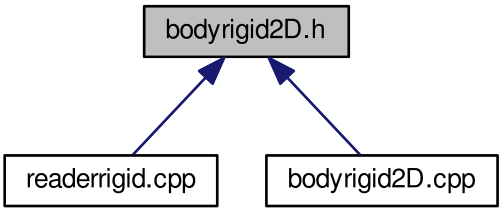
7.28 bodyrigid2D.h File Reference

```
#include "bodyrigid.h"
```

Include dependency graph for bodyrigid2D.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::RigidBody2D

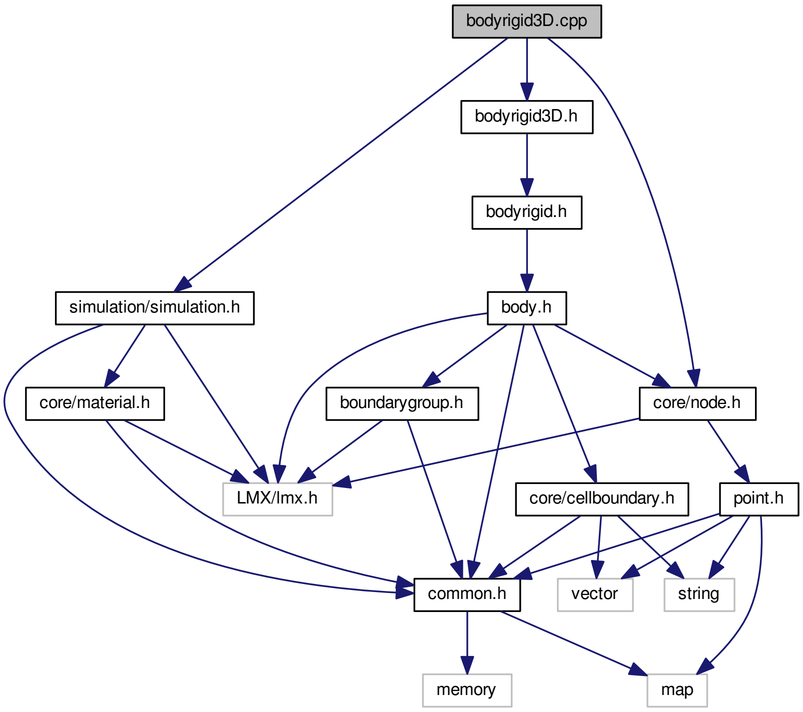
Namespaces

- mknix

7.29 bodyrigid3D.cpp File Reference

```
#include "bodyrigid3D.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for bodyrigid3D.cpp:



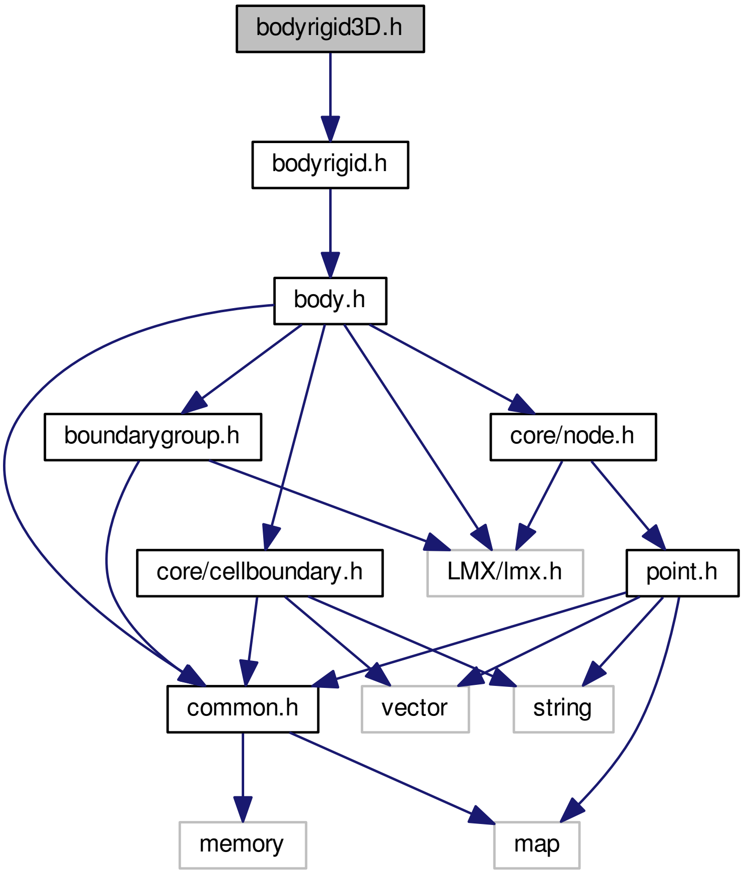
Namespaces

- mknix

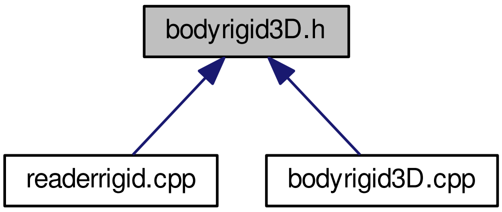
7.30 bodyrigid3D.h File Reference

```
#include "bodyrigid.h"
```

Include dependency graph for bodyrigid3D.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::RigidBody3D

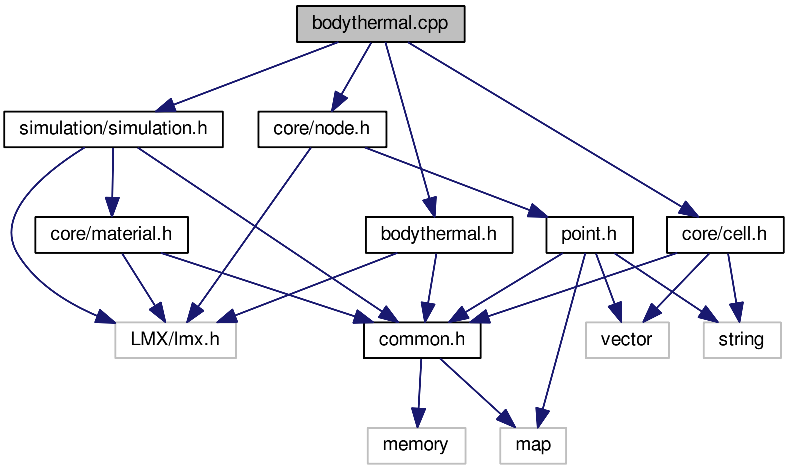
Namespaces

- 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:



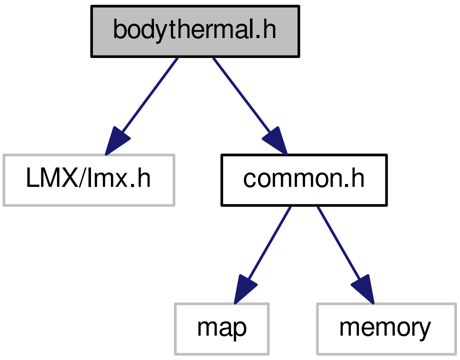
Namespaces

- mknix

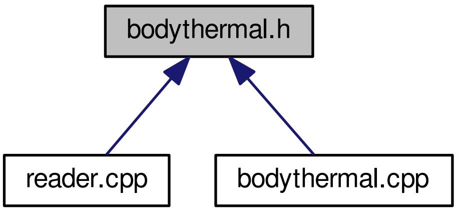
7.32 bodythermal.h File Reference

```
#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

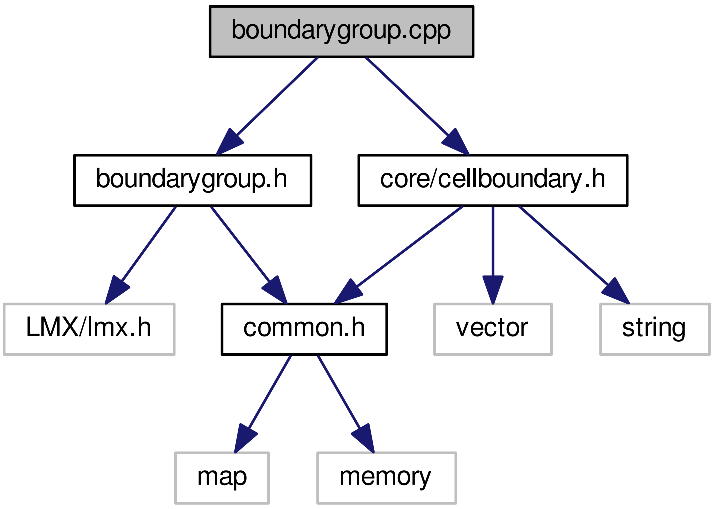
Namespaces

- mknix

7.33 boundarygroup.cpp File Reference

```
#include "boundarygroup.h"
#include <core/cellboundary.h>
```

Include dependency graph for boundarygroup.cpp:



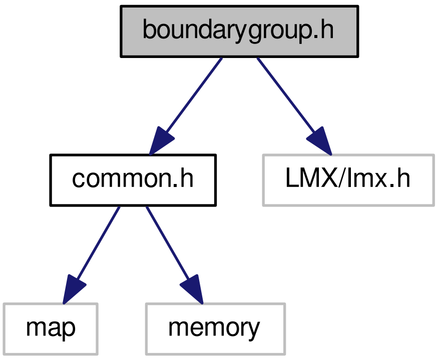
Namespaces

- mknix

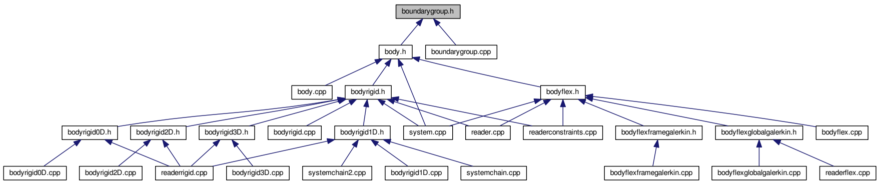
7.34 boundarygroup.h File Reference

```
#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

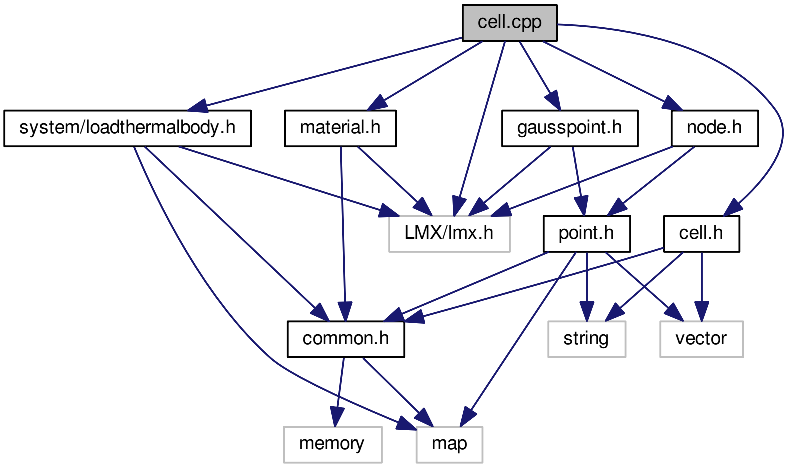
Namespaces

- 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

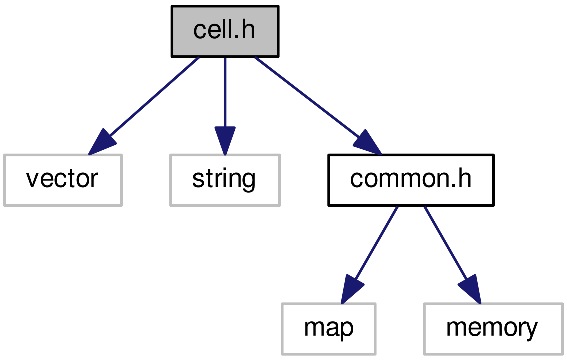
- mknix

7.36 cell.h File Reference

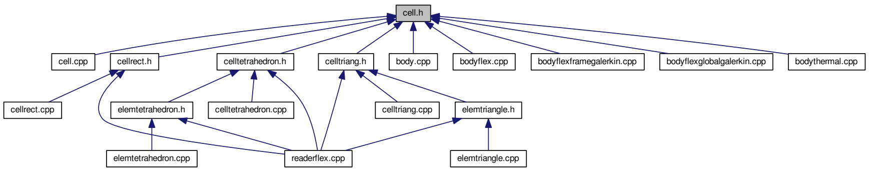
Background cells for integration.

```
#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 lmx::Vector< T >
- class lmx::Matrix< T >
- class mknix::Cell

Namespaces

- lmx
- mknix

7.36.1 Detailed Description

Background cells for integration.

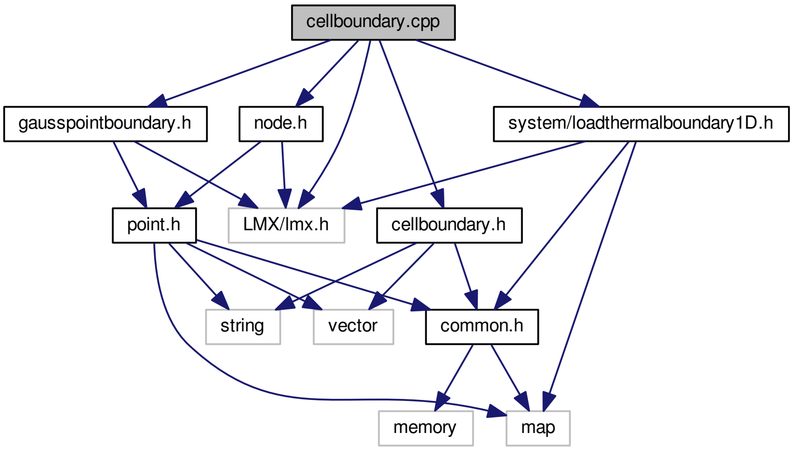
Author

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:



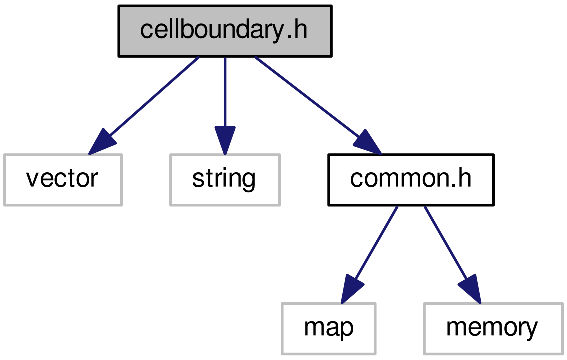
Namespaces

- mknix

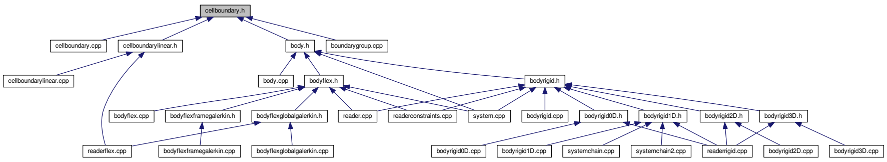
7.38 cellboundary.h File Reference

```
#include <vector>
#include <string>
#include "common.h"
```

Include dependency graph for cellboundary.h:



This graph shows which files directly or indirectly include this file:



Classes

- class lmx::Vector< T >
- class lmx::Matrix< T >
- class mknix::CellBoundary

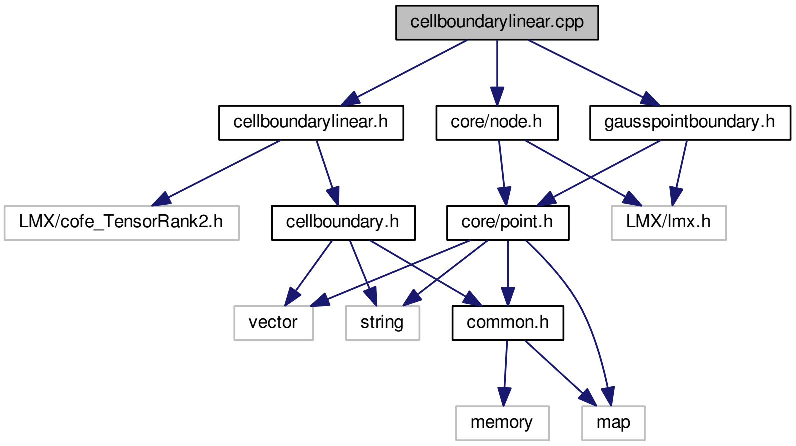
Namespaces

- lmx
- mknix

7.39 cellboundarylinear.cpp File Reference

```
#include "cellboundarylinear.h"
#include "gausspointboundary.h"
#include <core/node.h>
```

Include dependency graph for cellboundarylinear.cpp:



Namespaces

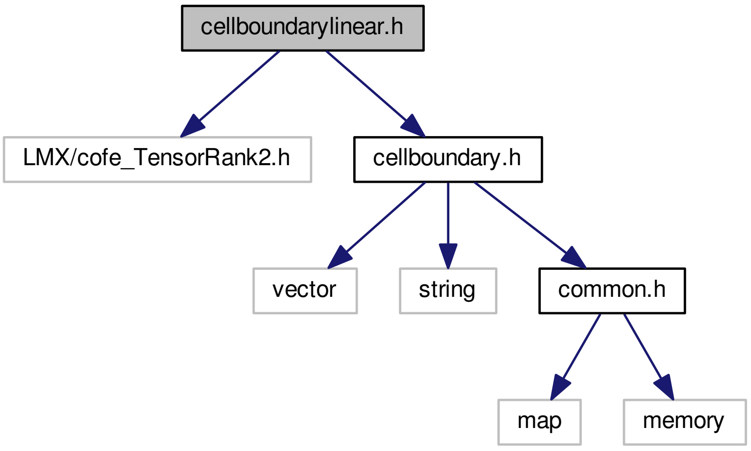
- mknix

7.40 cellboundarylinear.h File Reference

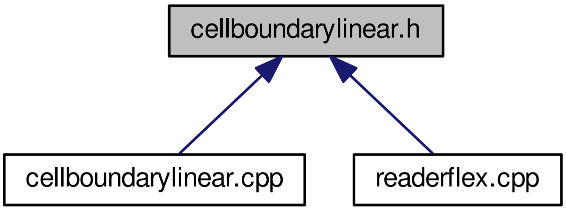
Background cells for boundary integration.

```
#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.

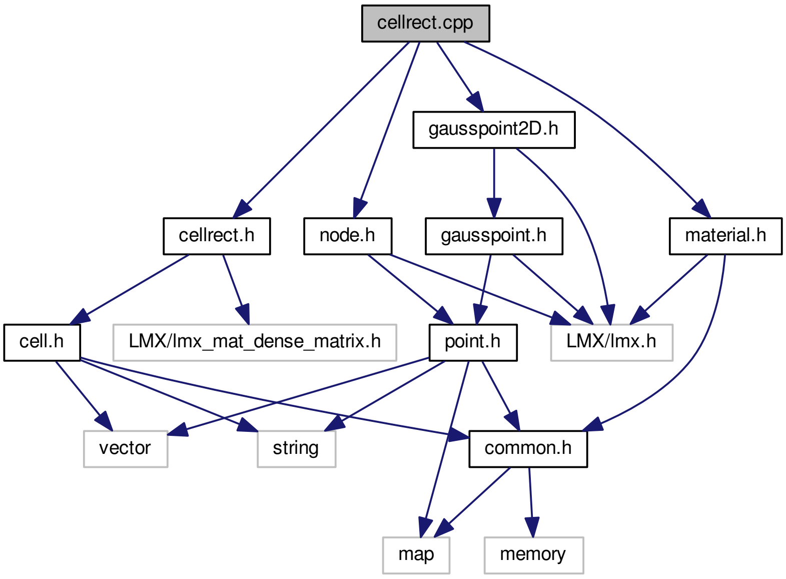
Author

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

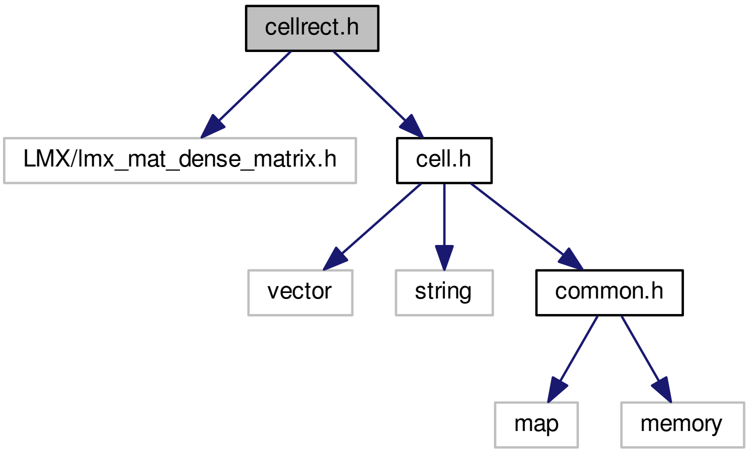
- mknix

7.42 cellrect.h File Reference

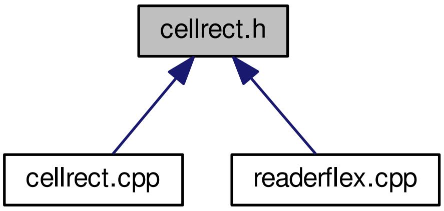
Background cells for integration.

```
#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 mknix::CellRect

Namespaces

- lmx
- mknix

7.42.1 Detailed Description

Background cells for integration.

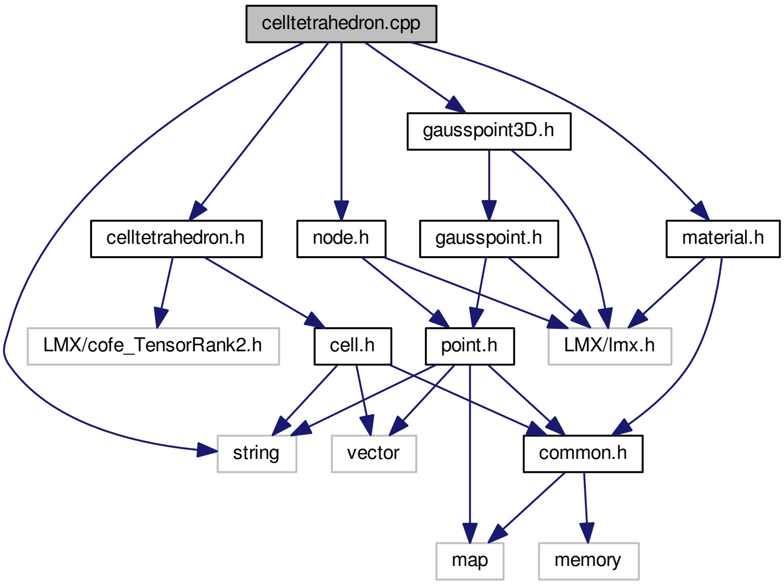
Author

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

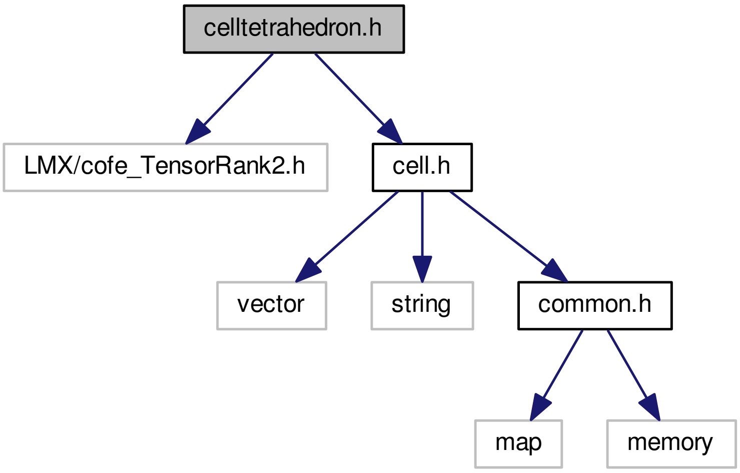
- mknix

7.44 celltetrahedron.h File Reference

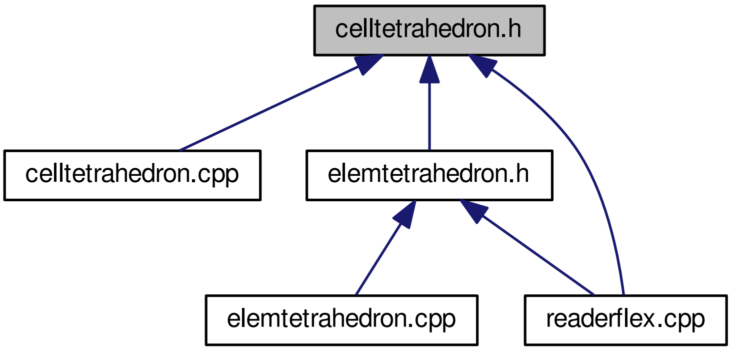
Background cells for integration.

```
#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 mknix::CellTetrahedron

Namespaces

- mknix

7.44.1 Detailed Description

Background cells for integration.

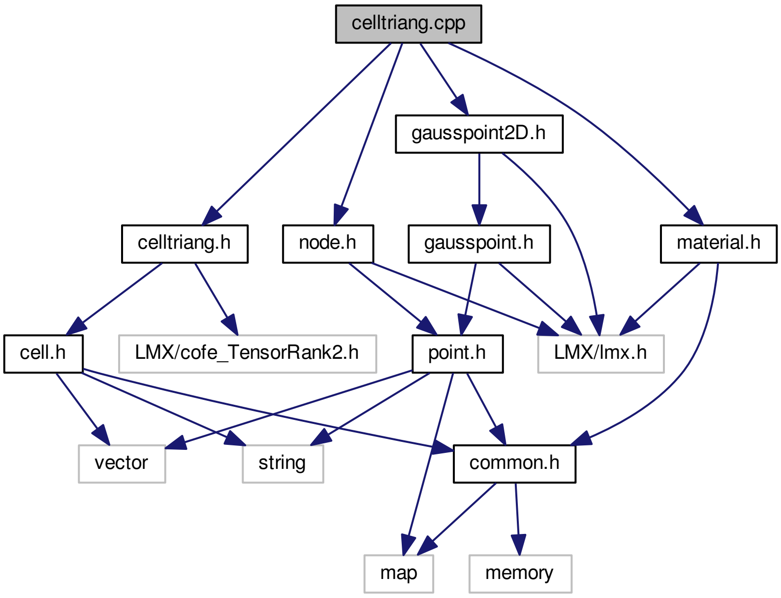
Author

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

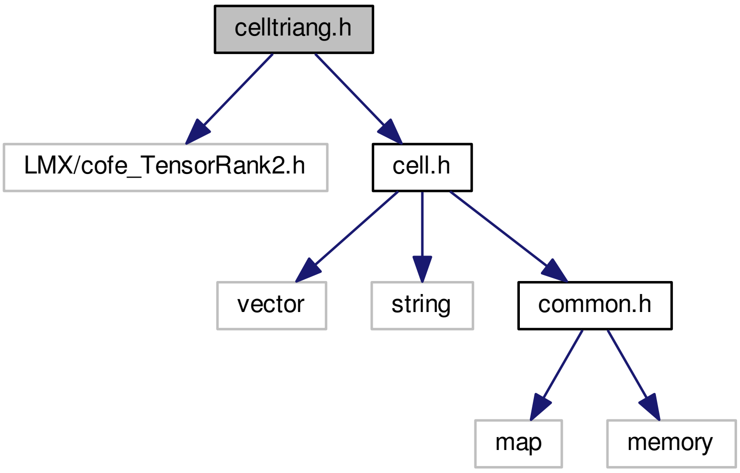
- mknix

7.46 celltriang.h File Reference

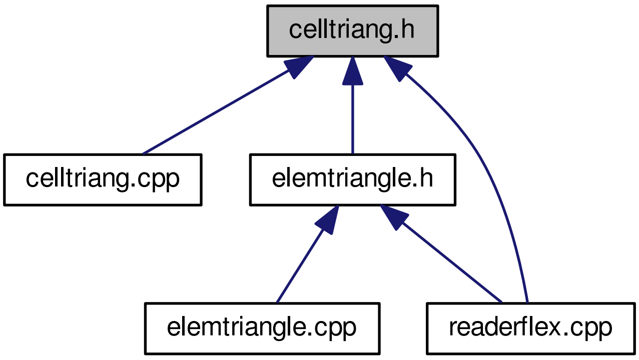
Background cells for integration.

```
#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 mknix::CellTriang

Namespaces

- mknix

7.46.1 Detailed Description

Background cells for integration.

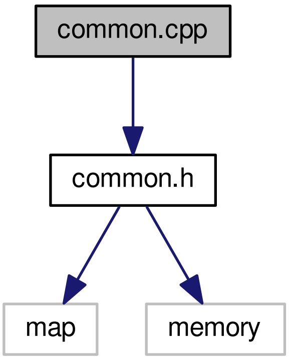
Author

Daniel Iglesias

7.47 common.cpp File Reference

```
#include "common.h"
```

Include dependency graph for common.cpp:



Namespaces

- mknix

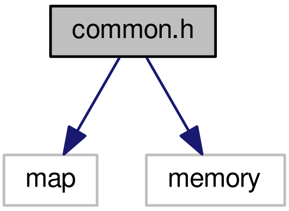
Functions

- double mknix::interpolate1D (double key, const std::map< double, double > &theMap)

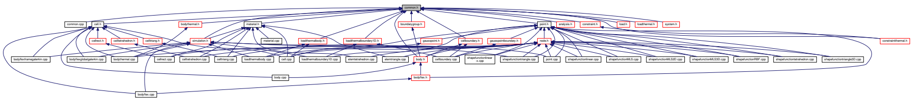
7.48 common.h File Reference

```
#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

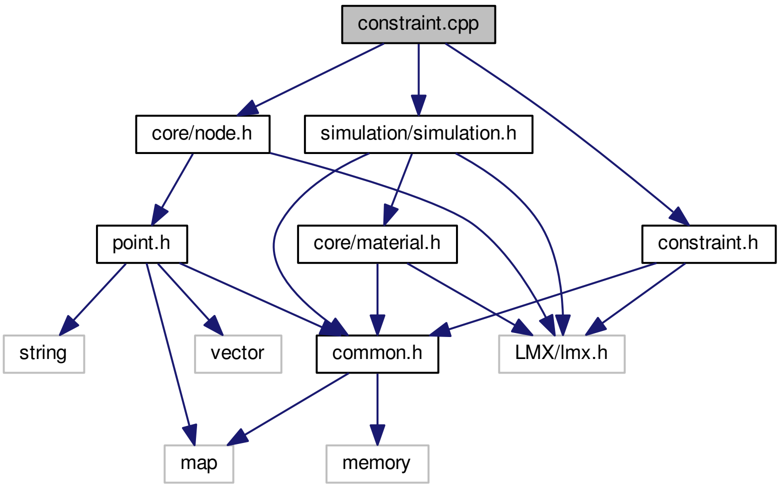
Namespaces

- mknix

7.51 constraint.cpp File Reference

```
#include "constraint.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for constraint.cpp:



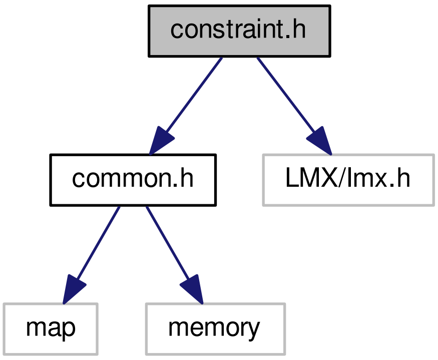
Namespaces

- mknix

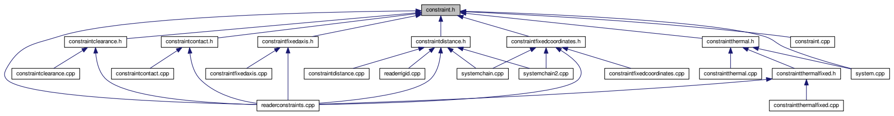
7.52 constraint.h File Reference

```
#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

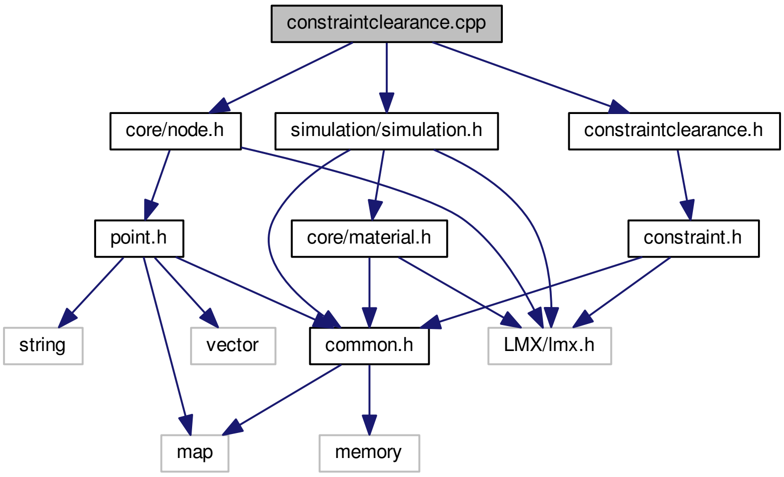
Namespaces

- mknix

7.53 constraintclearance.cpp File Reference

```
#include "constraintclearance.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for constraintclearance.cpp:



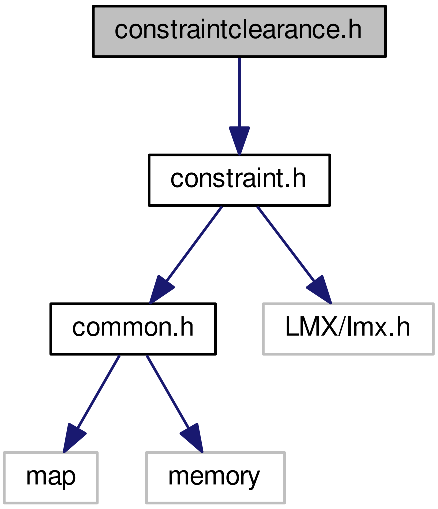
Namespaces

- mknix

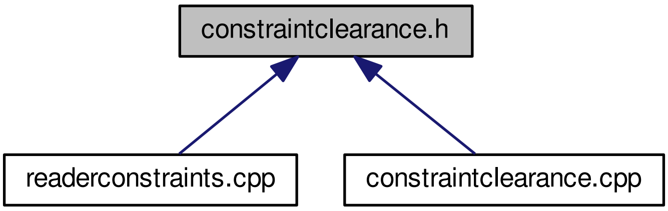
7.54 constraintclearance.h File Reference

```
#include "constraint.h"
```

Include dependency graph for constraintclearance.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ConstraintClearance

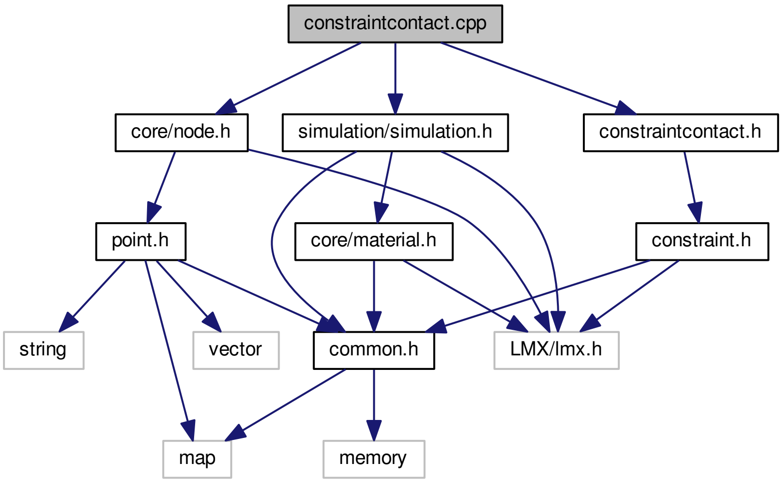
Namespaces

- mknix

7.55 constraintcontact.cpp File Reference

```
#include "constraintcontact.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for constraintcontact.cpp:



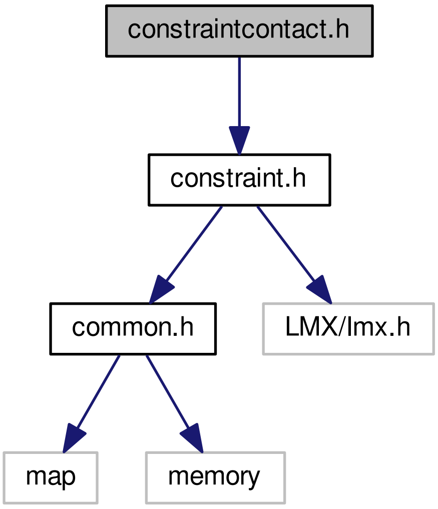
Namespaces

- mknix

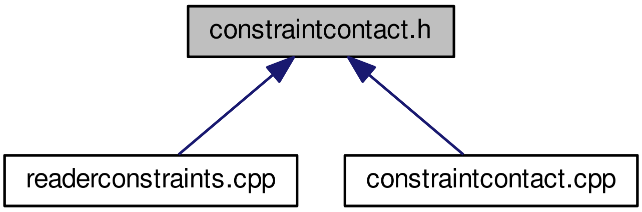
7.56 constraintcontact.h File Reference

```
#include "constraint.h"
```

Include dependency graph for constraintcontact.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ConstraintContact

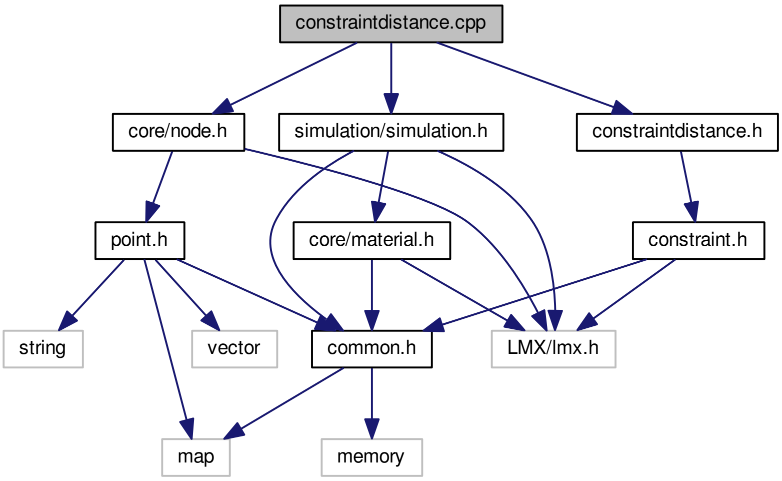
Namespaces

- mknix

7.57 constraintdistance.cpp File Reference

```
#include "constraintdistance.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for constraintdistance.cpp:



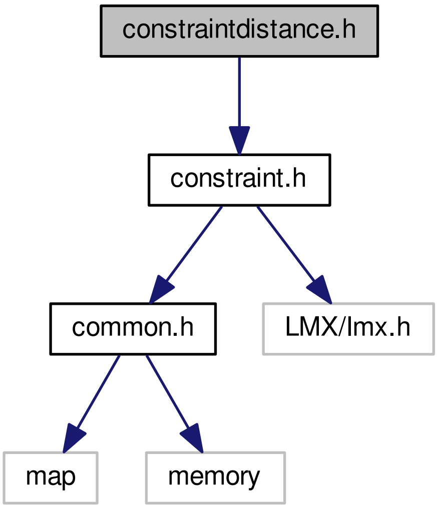
Namespaces

- mknix

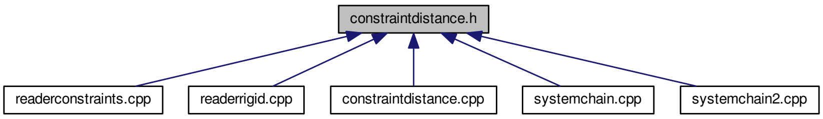
7.58 constraintdistance.h File Reference

```
#include "constraint.h"
```

Include dependency graph for constraintdistance.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ConstraintDistance

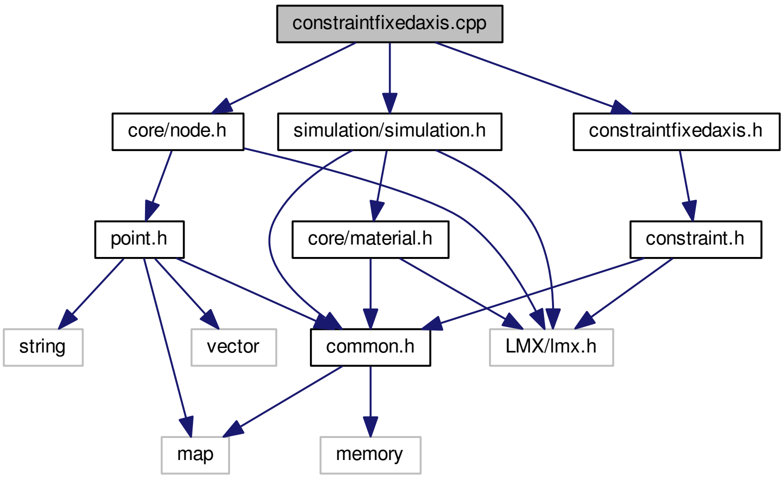
Namespaces

- mknix

7.59 constraintfixedaxis.cpp File Reference

```
#include "constraintfixedaxis.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for constraintfixedaxis.cpp:



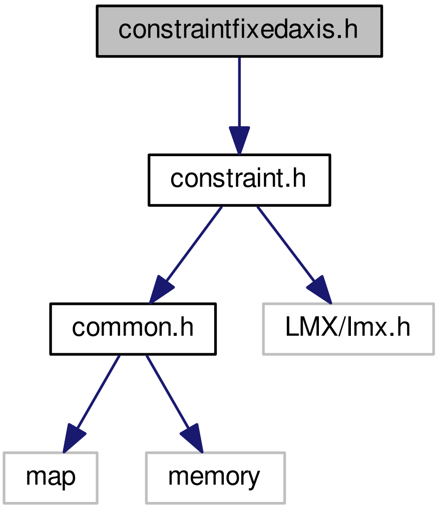
Namespaces

- mknix

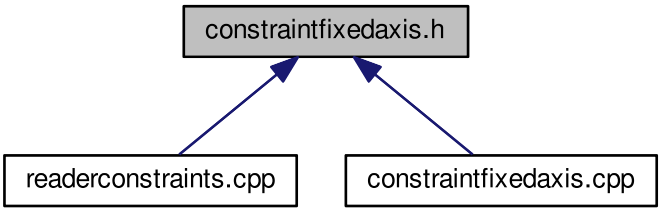
7.60 constraintfixedaxis.h File Reference

```
#include "constraint.h"
```

Include dependency graph for constraintfixedaxis.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ConstraintFixedAxis

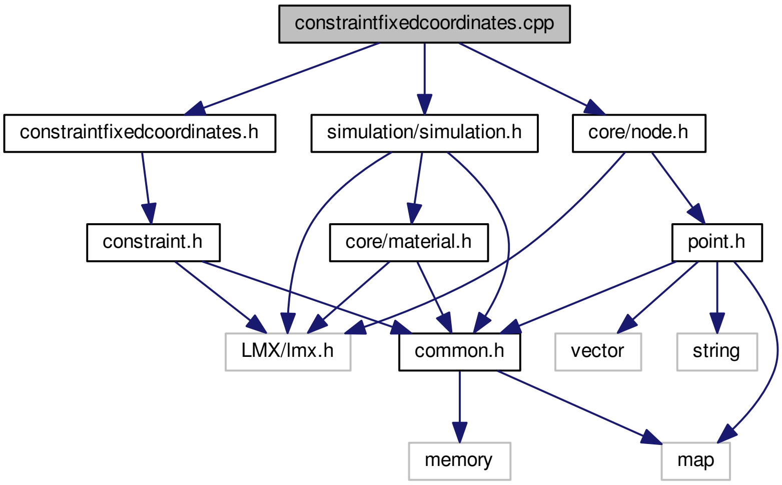
Namespaces

- mknix

7.61 constraintfixedcoordinates.cpp File Reference

```
#include "constraintfixedcoordinates.h"
#include <simulation/simulation.h>
#include <core/node.h>
```

Include dependency graph for constraintfixedcoordinates.cpp:



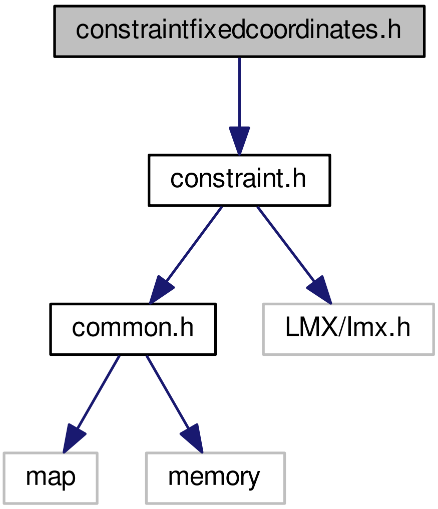
Namespaces

- mknix

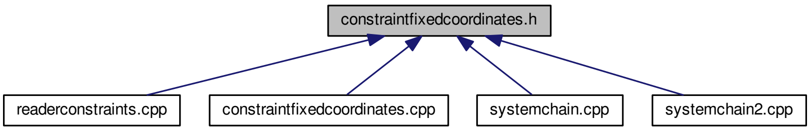
7.62 constraintfixedcoordinates.h File Reference

```
#include "constraint.h"
```

Include dependency graph for constraintfixedcoordinates.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ConstraintFixedCoordinates

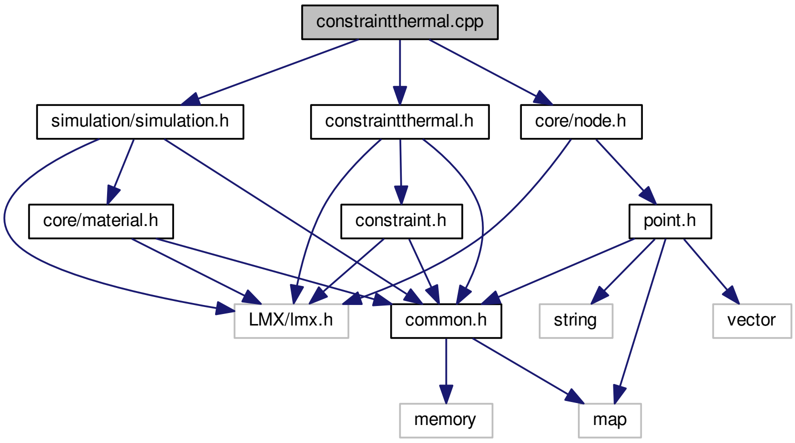
Namespaces

- mknix

7.63 constraintthermal.cpp File Reference

```
#include "constraintthermal.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for constraintthermal.cpp:



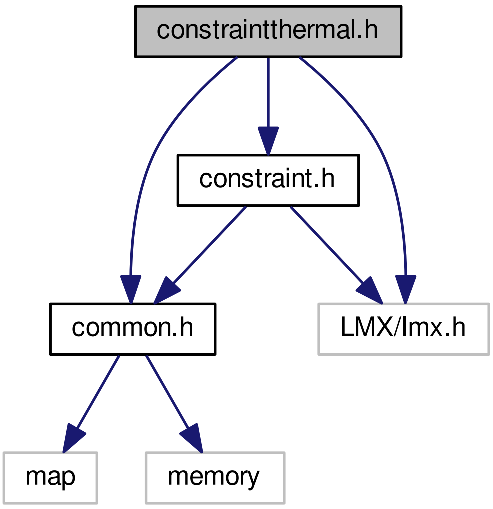
Namespaces

- mknix

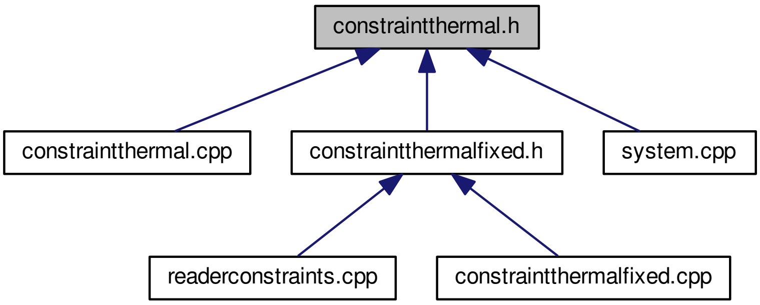
7.64 constraintthermal.h File Reference

```
#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 mknix::ConstraintThermal

Namespaces

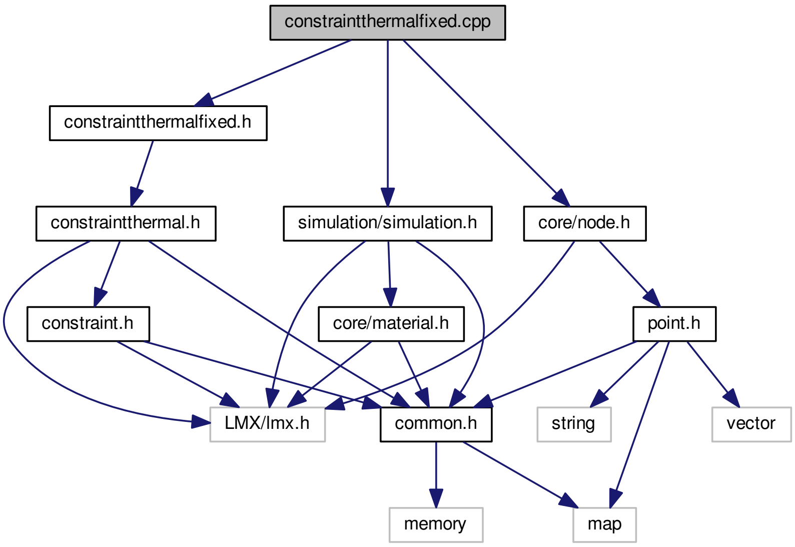
- mknix

7.65 constraintthermalfixed.cpp File Reference

```
#include "constraintthermalfixed.h"
```

```
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for constraintthermalfixed.cpp:



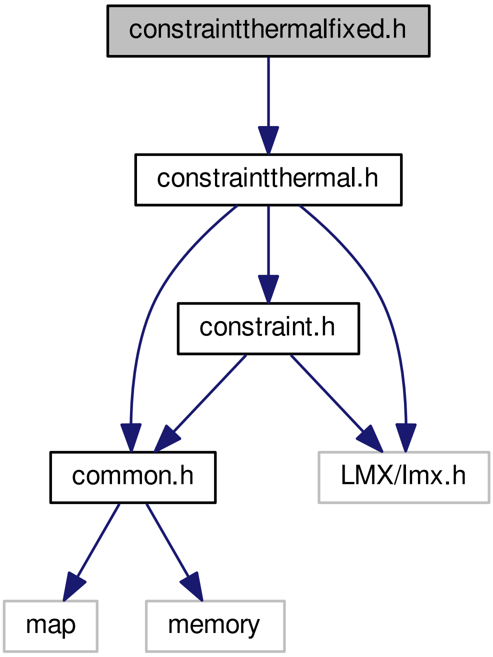
Namespaces

- mknix

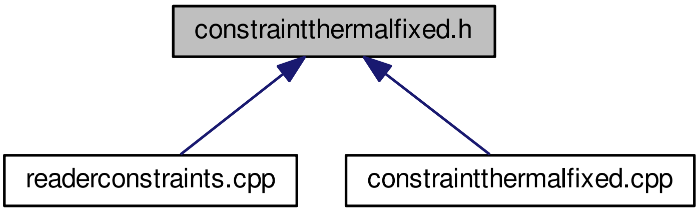
7.66 constraintthermalfixed.h File Reference

```
#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

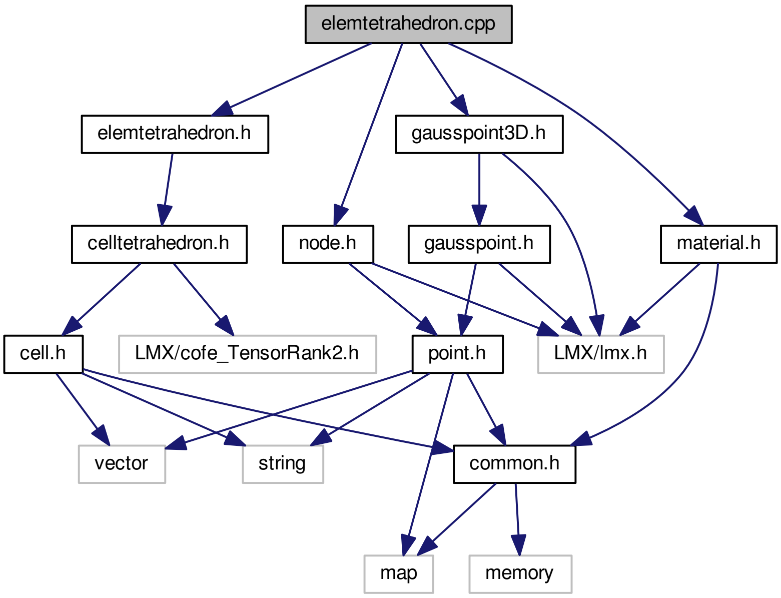
Namespaces

- 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:



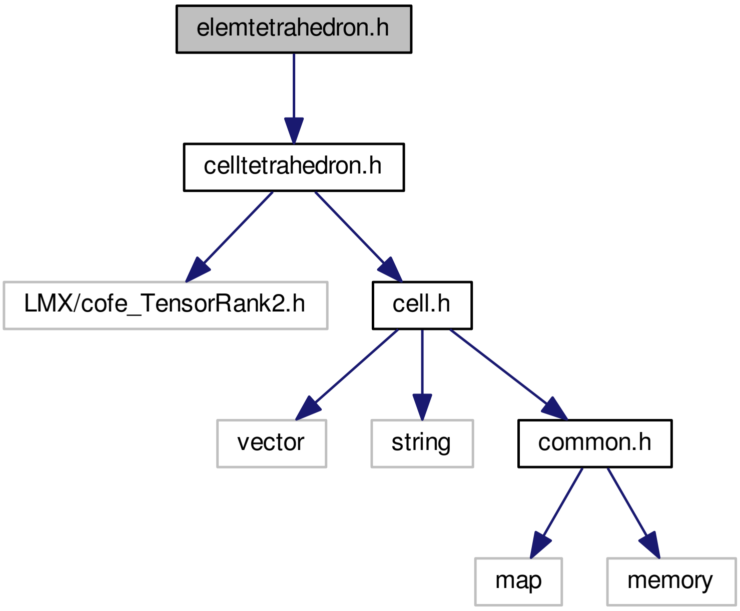
Namespaces

- mknix

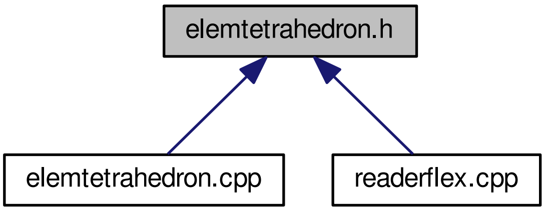
7.69 elemtetrahedron.h File Reference

```
#include "celltetrahedron.h"
```

Include dependency graph for elemtetrahedron.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ElemTetrahedron

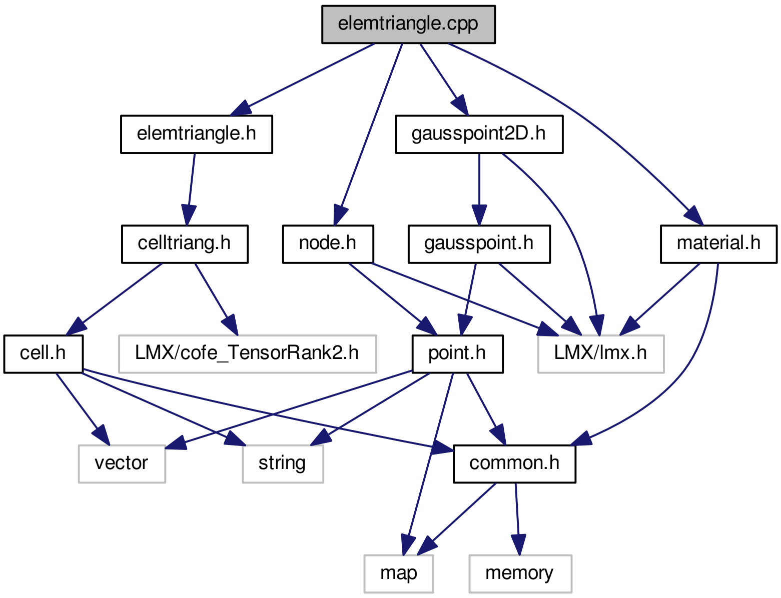
Namespaces

- mknix

7.70 elemtriangle.cpp File Reference

```
#include "elemtriangle.h"
#include "material.h"
#include "node.h"
#include "gausspoint2D.h"
```

Include dependency graph for elemtriangle.cpp:



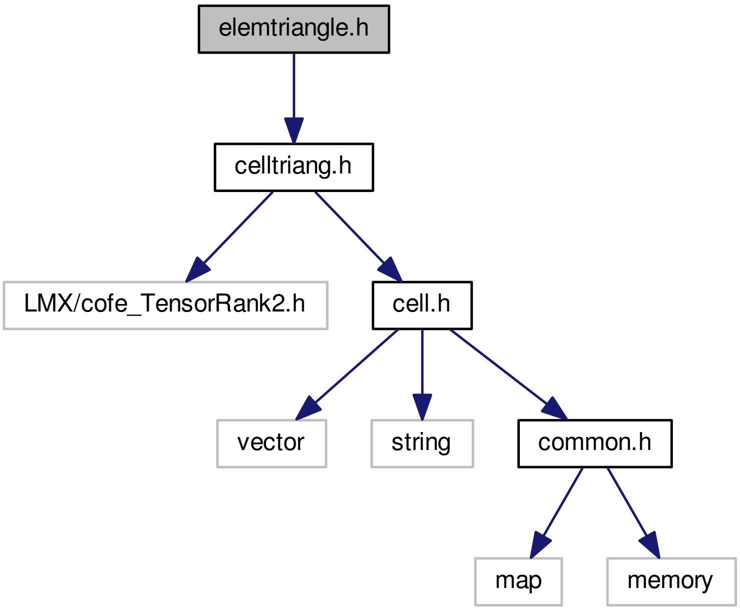
Namespaces

- mknix

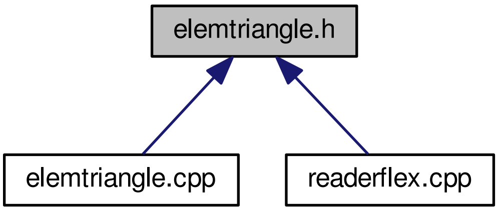
7.71 elemtriangle.h File Reference

```
#include "celltriang.h"
```

Include dependency graph for elemtriangle.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ElemTriangle

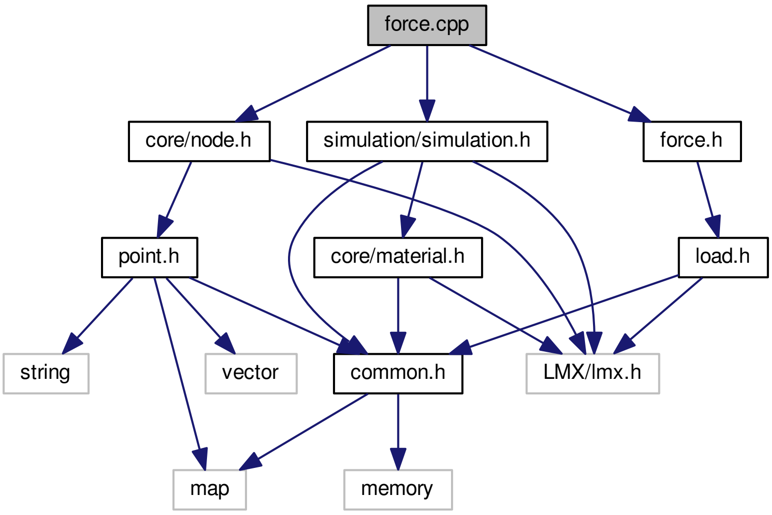
Namespaces

- mknix

7.72 force.cpp File Reference

```
#include "force.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for force.cpp:



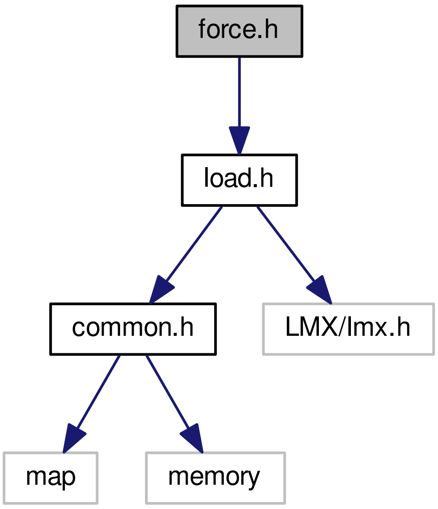
Namespaces

- mknix

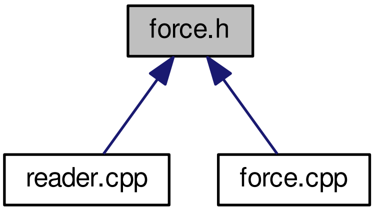
7.73 force.h File Reference

```
#include "load.h"
```

Include dependency graph for force.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::Force

Namespaces

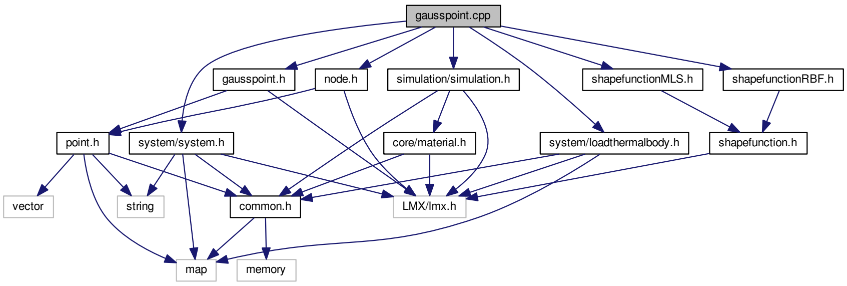
- mknix

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

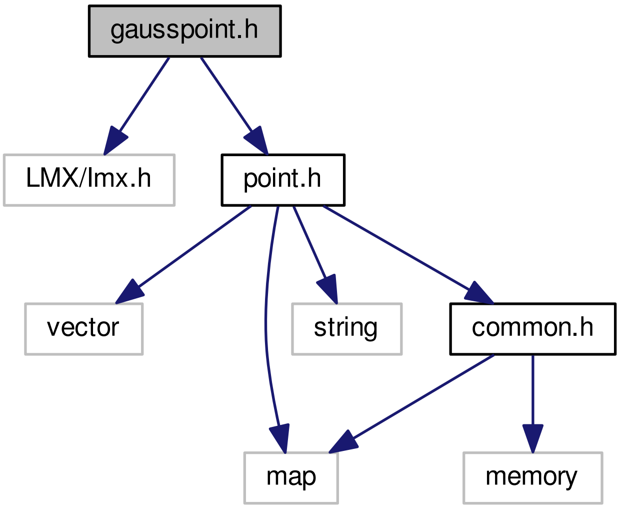
- mknix

7.75 gausspoint.h File Reference

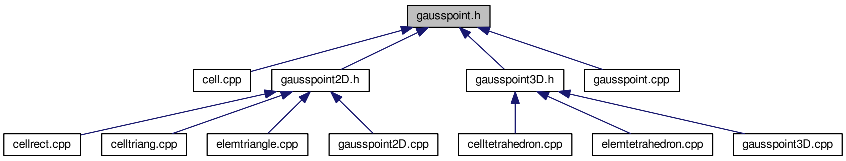
Point for numerical integration.

```
#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.

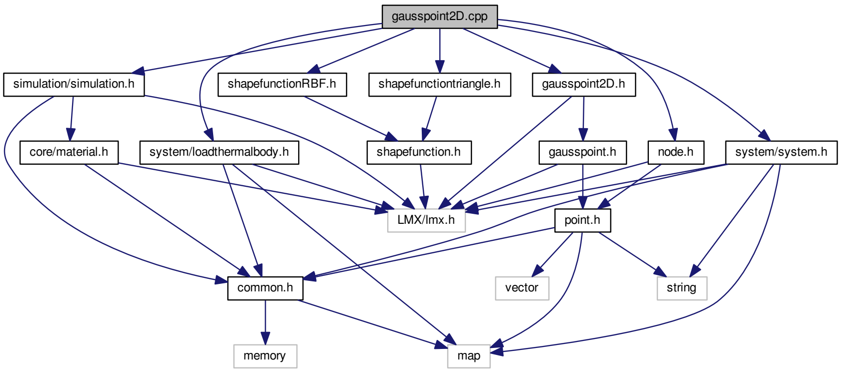
Author

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:



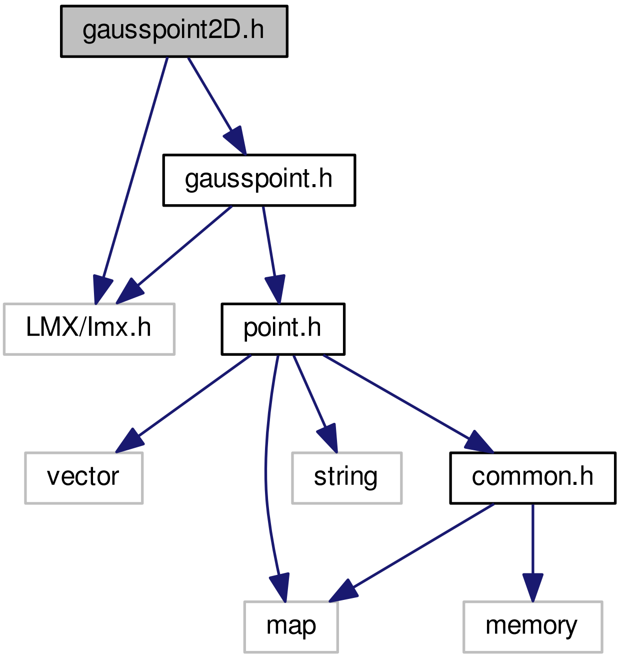
Namespaces

- mknix

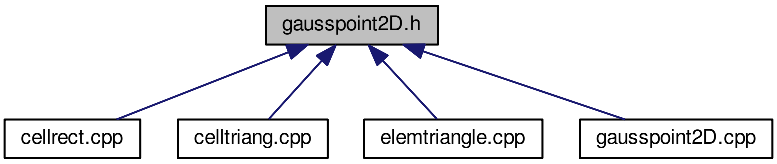
7.77 gausspoint2D.h File Reference

```
#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

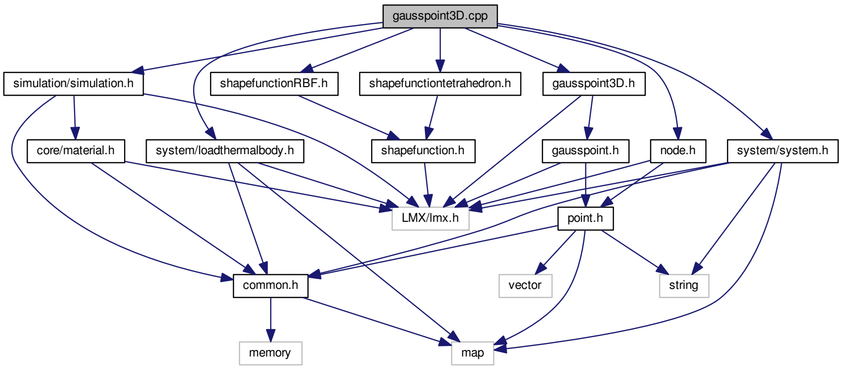
Namespaces

- 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:



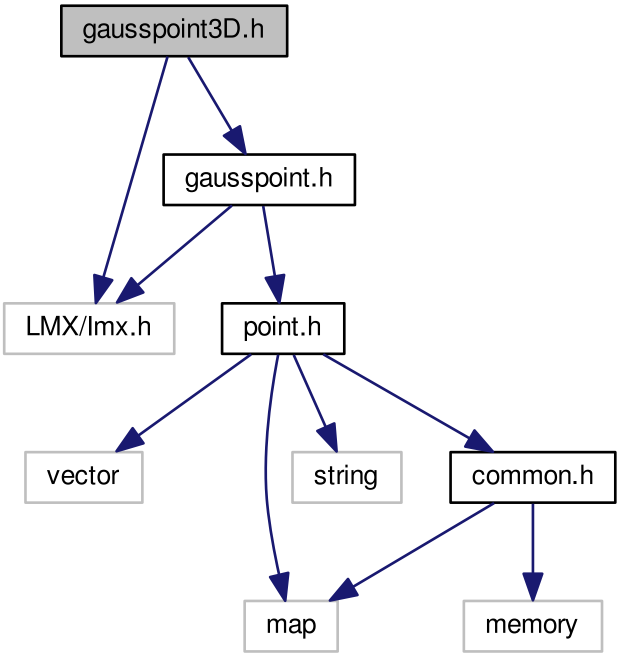
Namespaces

- mknix

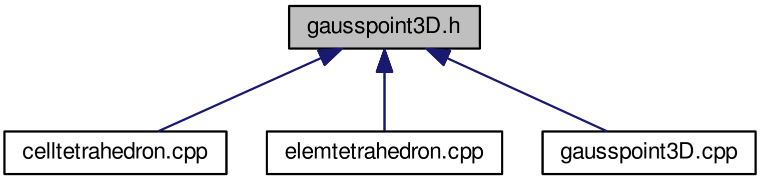
7.79 gausspoint3D.h File Reference

```
#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 mknix::GaussPoint3D

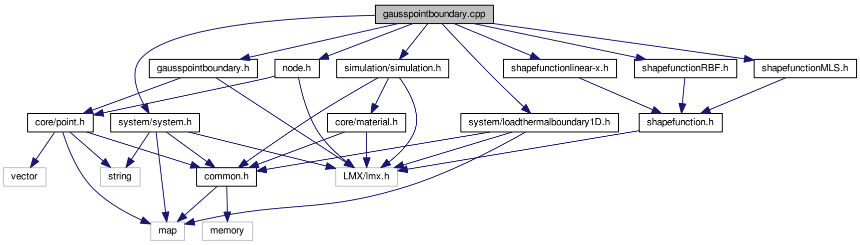
Namespaces

- mknix

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:



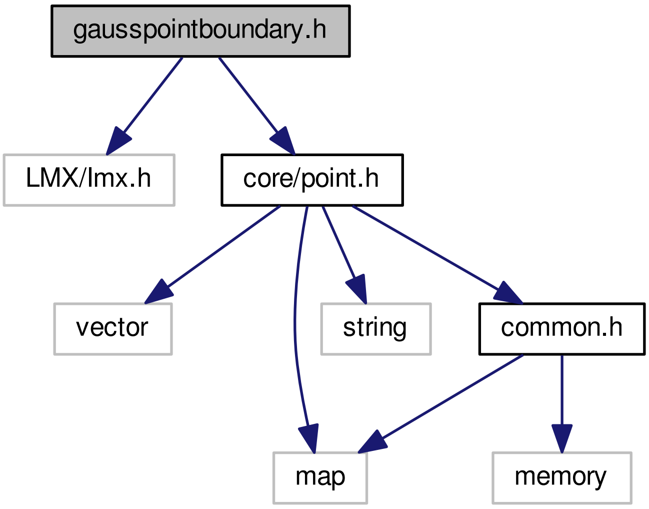
Namespaces

- mknix

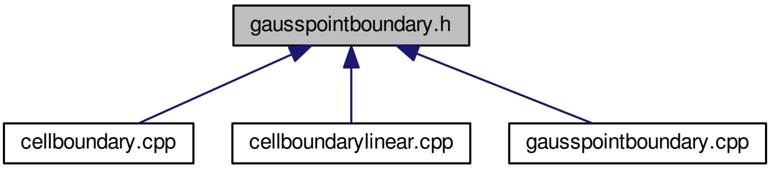
7.81 gausspointboundary.h File Reference

```
#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

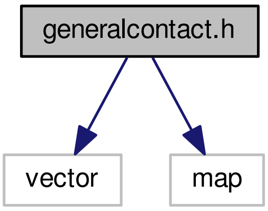
- mknix

7.82 generalcontact.cpp File Reference

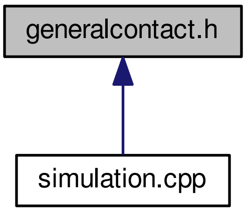
7.83 generalcontact.h File Reference

```
#include <vector>
#include <map>
```

Include dependency graph for generalcontact.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::Contact

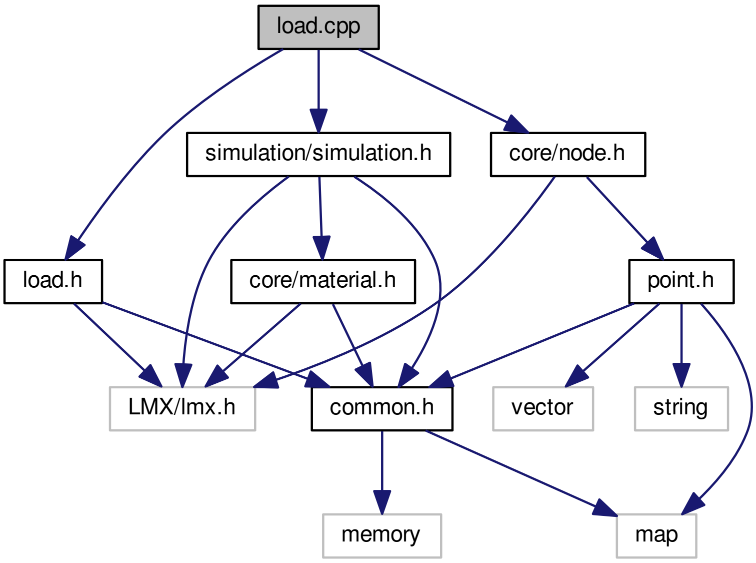
Namespaces

- mknix

7.84 load.cpp File Reference

```
#include "load.h"
#include <simulation/simulation.h>
#include <core/node.h>
```

Include dependency graph for load.cpp:



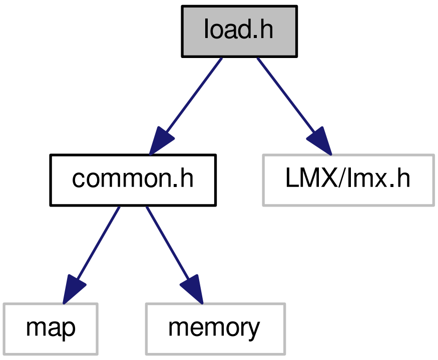
Namespaces

- mknix

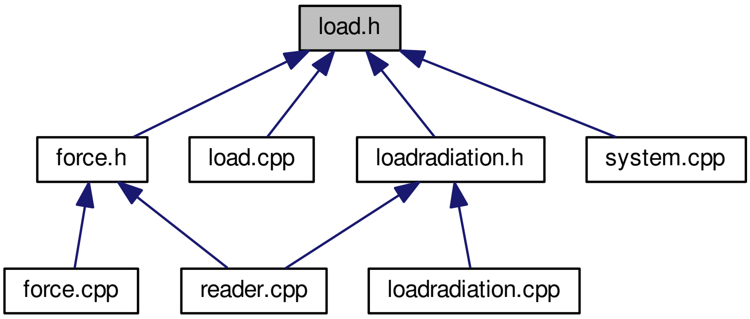
7.85 load.h File Reference

```
#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

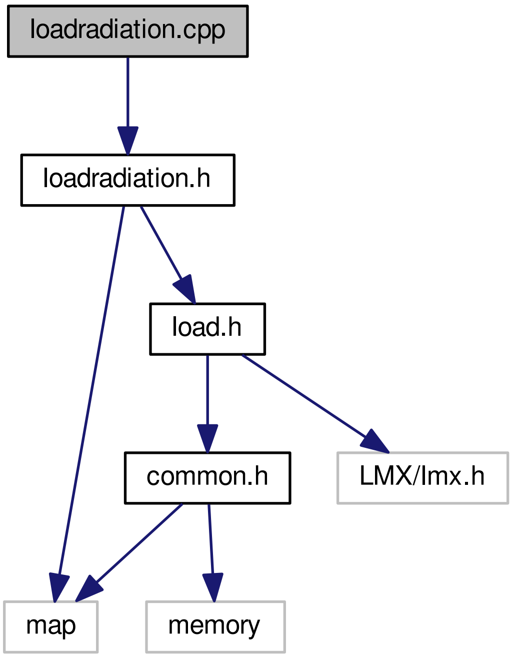
Namespaces

- mknix

7.86 loadradiation.cpp File Reference

```
#include "loadradiation.h"
```

Include dependency graph for loadradiation.cpp:



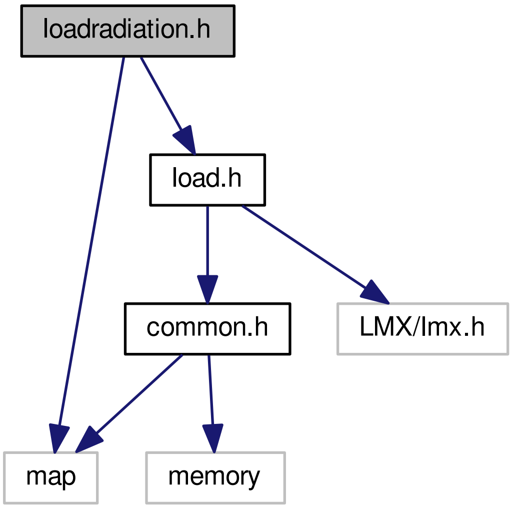
Namespaces

- mknix

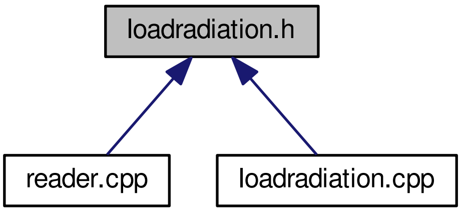
7.87 loadradiation.h File Reference

```
#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

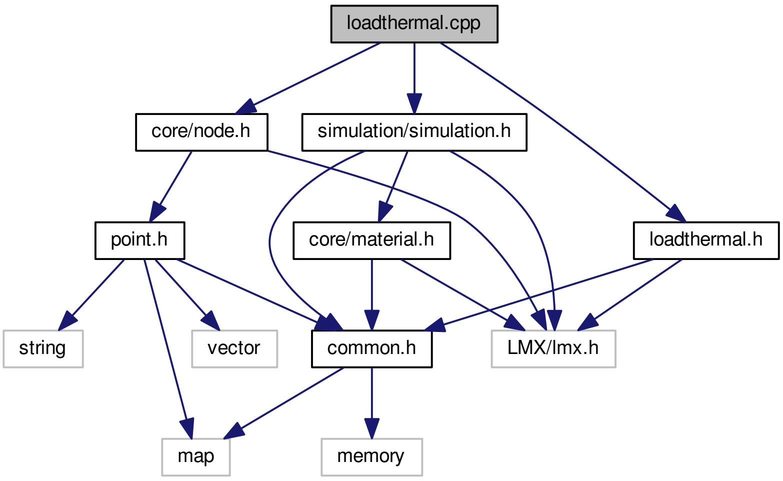
Namespaces

- mknix

7.88 loadthermal.cpp File Reference

```
#include "loadthermal.h"
#include <core/node.h>
#include <simulation/simulation.h>
```

Include dependency graph for loadthermal.cpp:



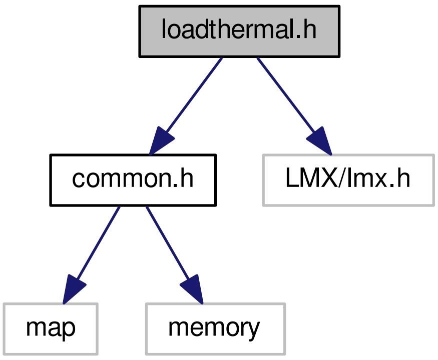
Namespaces

- mknix

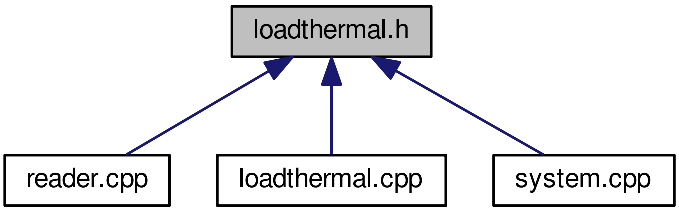
7.89 loadthermal.h File Reference

```
#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

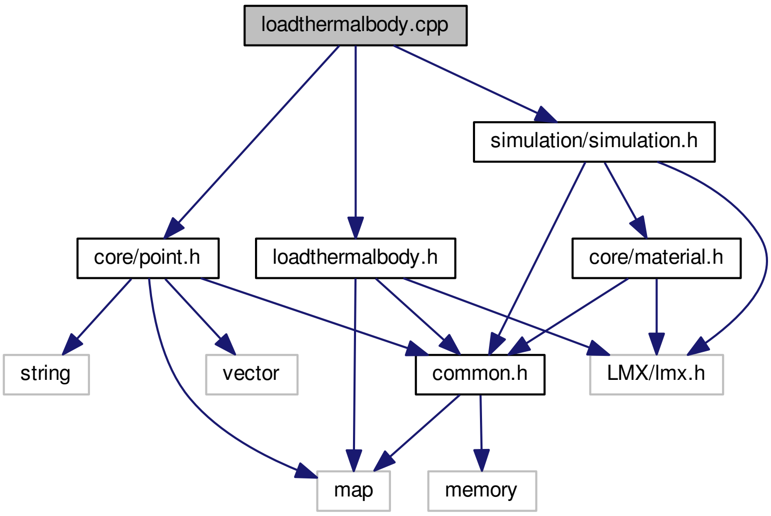
Namespaces

- mknix

7.90 loadthermalbody.cpp File Reference

```
#include "loadthermalbody.h"
#include <core/point.h>
#include <simulation/simulation.h>
```

Include dependency graph for loadthermalbody.cpp:



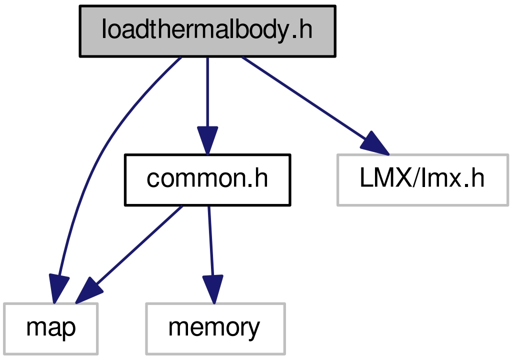
Namespaces

- mknix

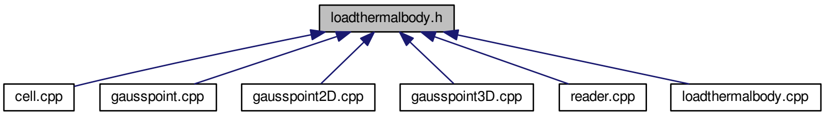
7.91 loadthermalbody.h File Reference

```
#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

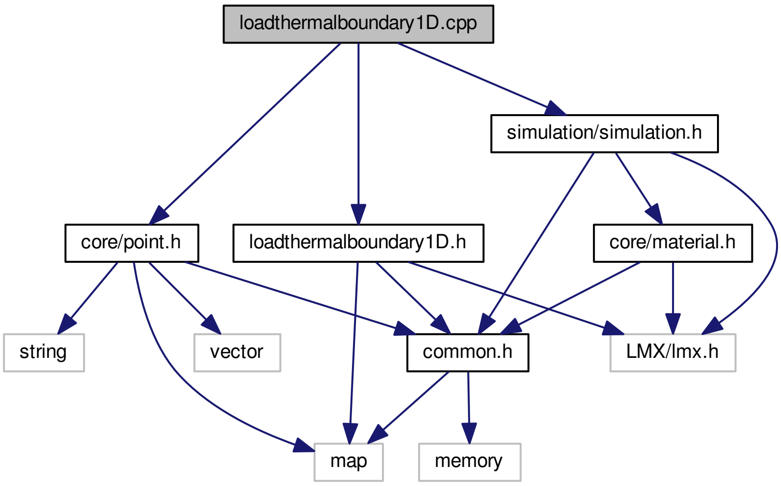
Namespaces

- mknix

7.92 loadthermalboundary1D.cpp File Reference

```
#include "loadthermalboundary1D.h"
#include <core/point.h>
#include <simulation/simulation.h>
```

Include dependency graph for loadthermalboundary1D.cpp:



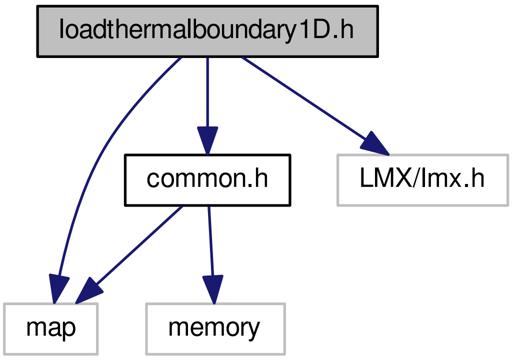
Namespaces

- mknix

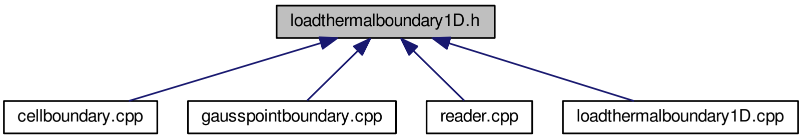
7.93 loadthermalboundary1D.h File Reference

```
#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

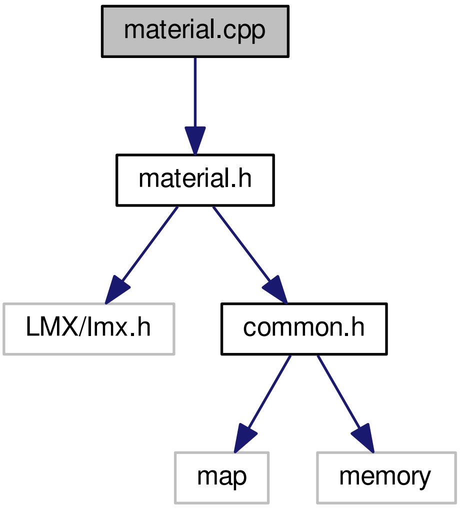
Namespaces

- mknix

7.94 material.cpp File Reference

```
#include "material.h"
```

Include dependency graph for material.cpp:



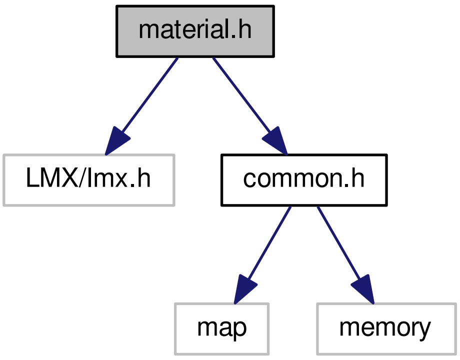
Namespaces

- 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

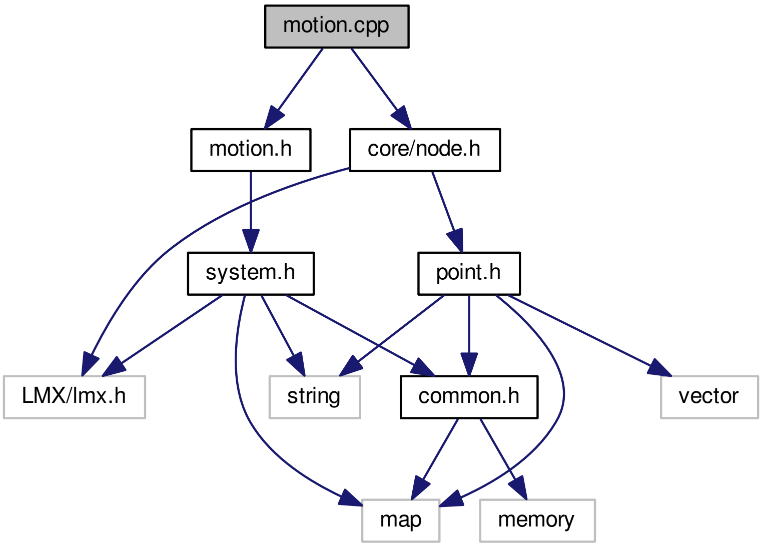
Namespaces

- mknix

7.96 motion.cpp File Reference

```
#include "motion.h"
#include <core/node.h>
```

Include dependency graph for motion.cpp:



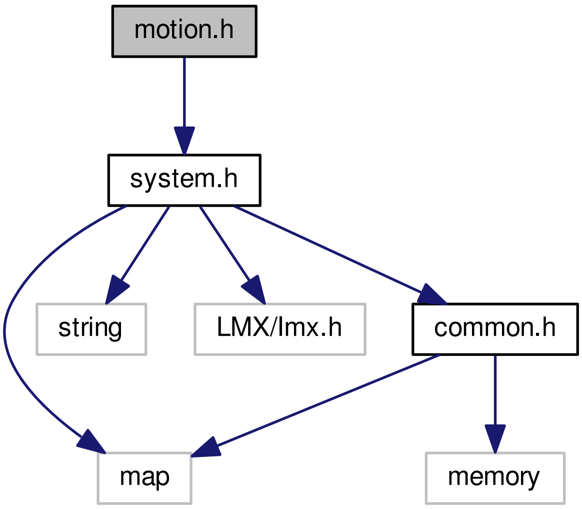
Namespaces

- mknix

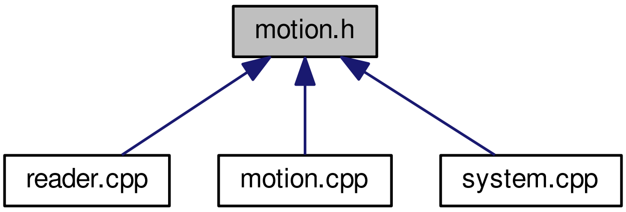
7.97 motion.h File Reference

```
#include "system.h"
```

Include dependency graph for motion.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::Motion

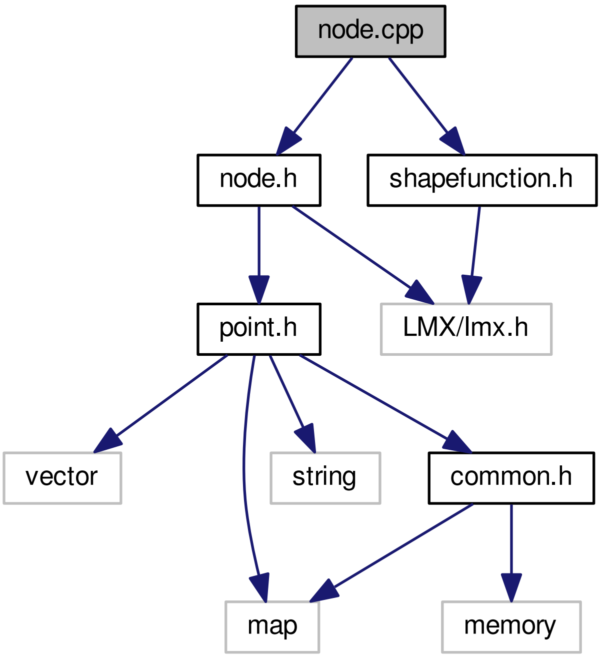
Namespaces

- mknix

7.98 node.cpp File Reference

```
#include "node.h"
#include "shapefunction.h"
```

Include dependency graph for node.cpp:



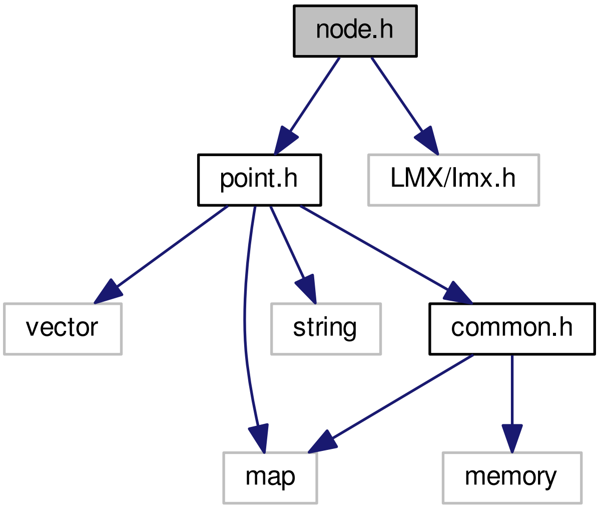
Namespaces

- 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

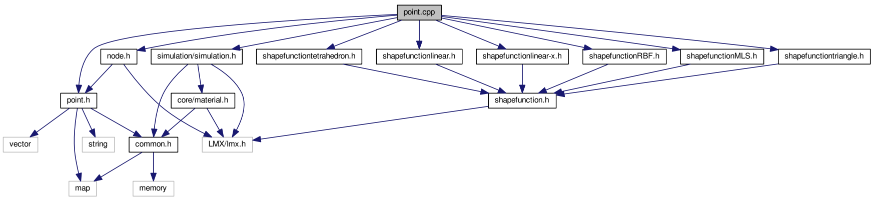
Namespaces

- 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

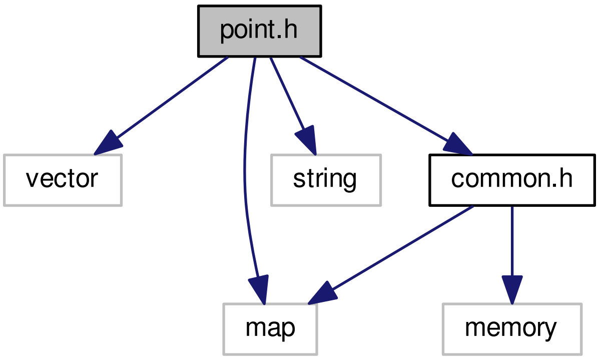
- mknix

7.101 point.h File Reference

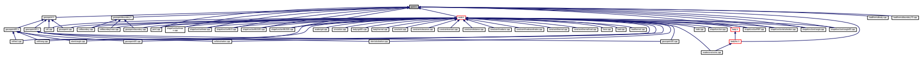
Point of interest.

```
#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.

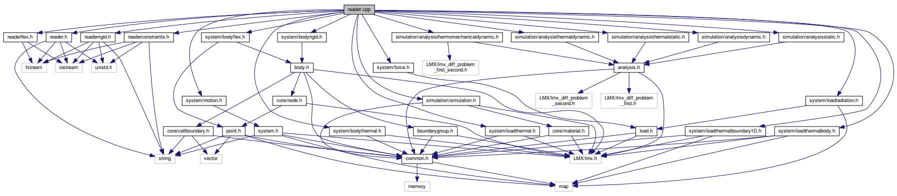
Author

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:



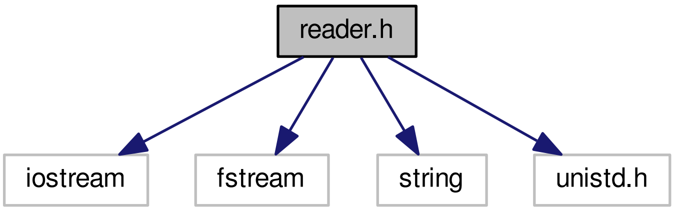
Namespaces

- mknix

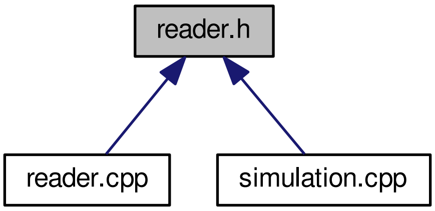
7.103 reader.h File Reference

```
#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 mknix::Reader

Namespaces

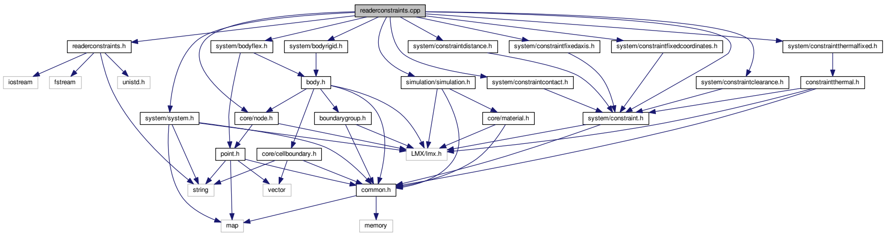
- mknix

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:



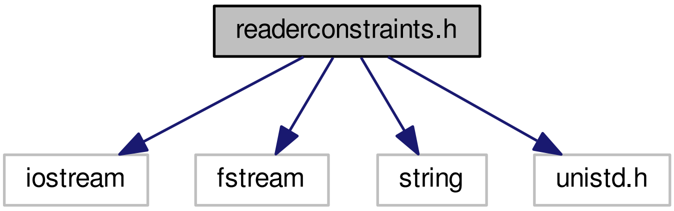
Namespaces

- mknix

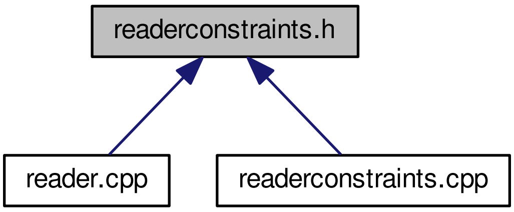
7.105 readerconstraints.h File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <unistd.h>
```

Include dependency graph for readerconstraints.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ReaderConstraints

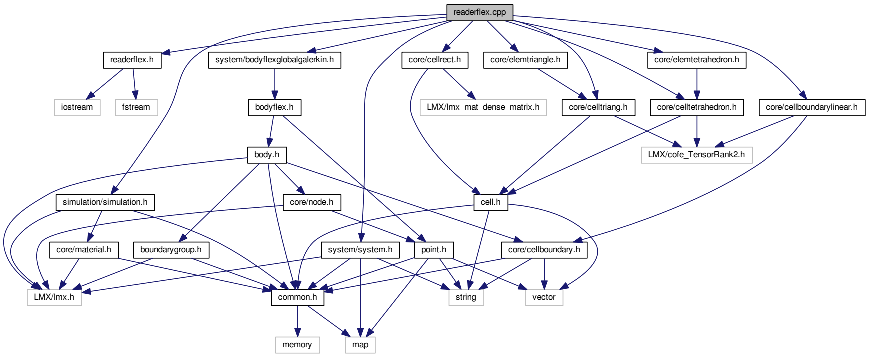
Namespaces

- mknix

7.106 readerflex.cpp File Reference

```
#include "readerflex.h"
#include <simulation/simulation.h>
#include <system/bodyflexglobalgalerkin.h>
#include <core/cellrect.h>
#include <core/celltriang.h>
#include <core/celltetrahedron.h>
#include <core/cellboundarylinear.h>
#include <core/elemtriangle.h>
#include <core/elemtetrahedron.h>
#include <system/system.h>
```

Include dependency graph for readerflex.cpp:



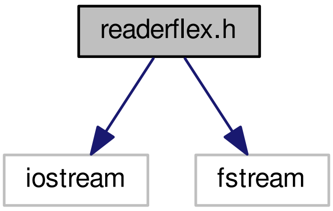
Namespaces

- mknix

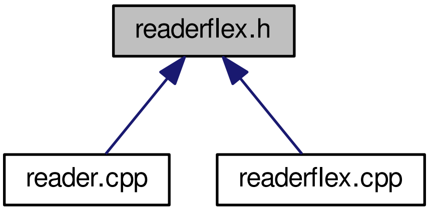
7.107 readerflex.h File Reference

```
#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

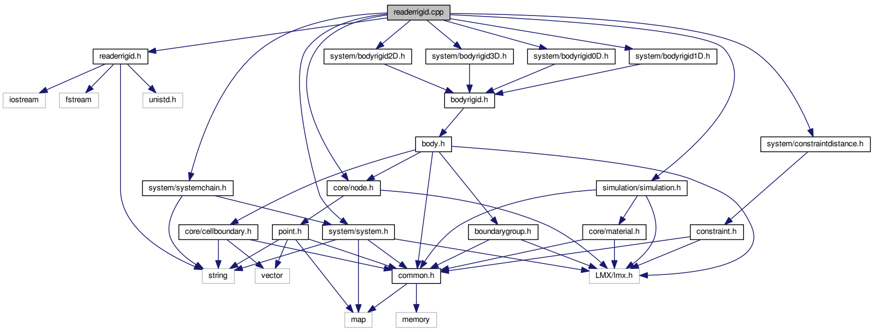
- mknix

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:



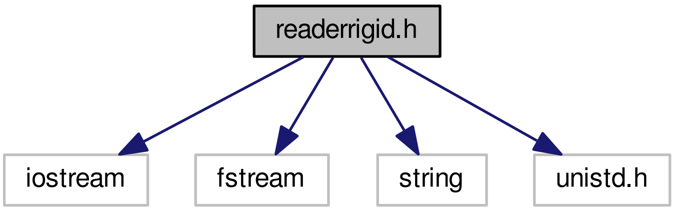
Namespaces

- mknix

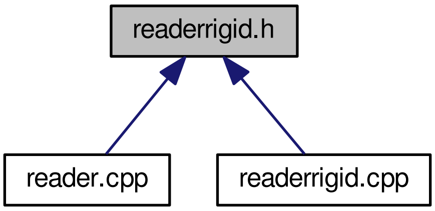
7.109 readerrigid.h File Reference

```
#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

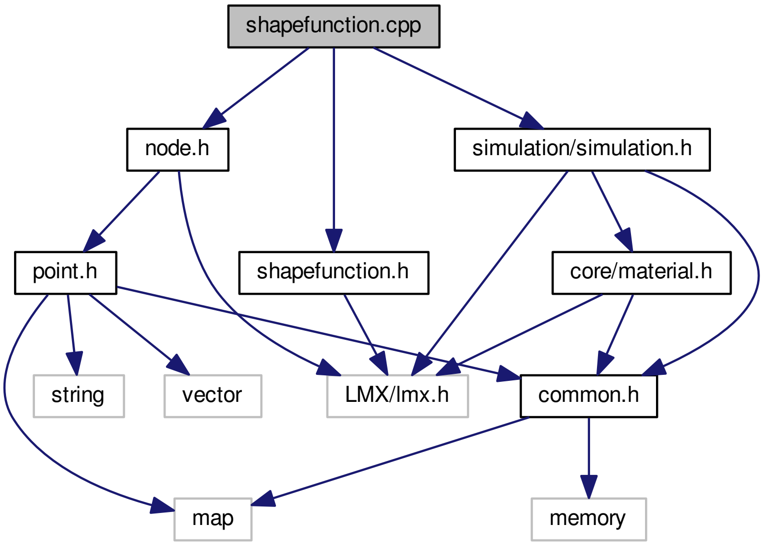
Namespaces

- mknix

7.110 shapefunction.cpp File Reference

```
#include "shapefunction.h"
#include "node.h"
#include <simulation/simulation.h>
```

Include dependency graph for shapefunction.cpp:



Namespaces

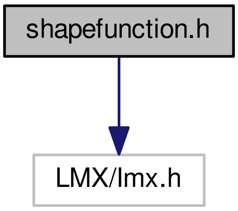
- mknix

7.111 shapefunction.h File Reference

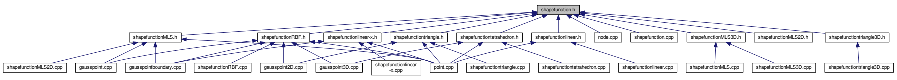
Function for interpolation of basic variables.

```
#include "LMX/lmx.h"
```

Include dependency graph for shapefunction.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ShapeFunction

Namespaces

- mknix

7.111.1 Detailed Description

Function for interpolation of basic variables.

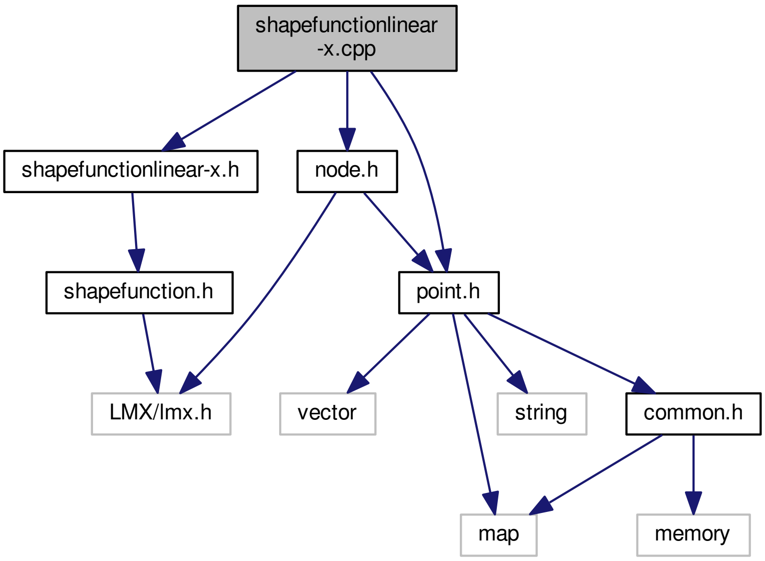
Author

Daniel Iglesias

7.112 shapefunctionlinear-x.cpp File Reference

```
#include "shapefunctionlinear-x.h"
#include "point.h"
#include "node.h"
```

Include dependency graph for shapefunctionlinear-x.cpp:



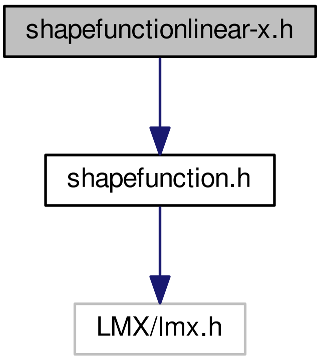
Namespaces

- mknix

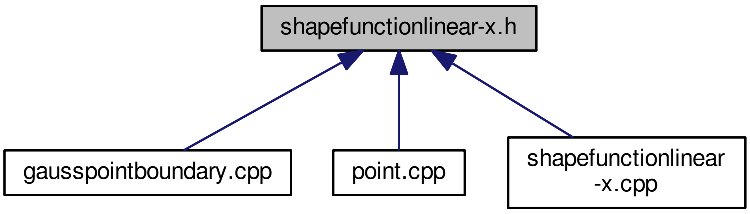
7.113 shapefunctionlinear-x.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionlinear-x.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ShapeFunctionLinearX

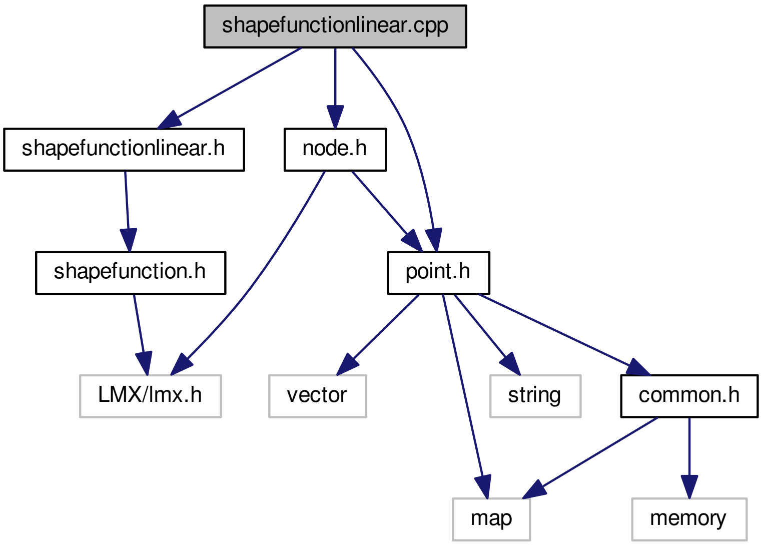
Namespaces

- mknix

7.114 shapefunctionlinear.cpp File Reference

```
#include "shapefunctionlinear.h"
#include "point.h"
#include "node.h"
```

Include dependency graph for shapefunctionlinear.cpp:



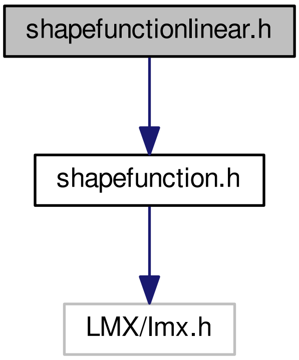
Namespaces

- mknix

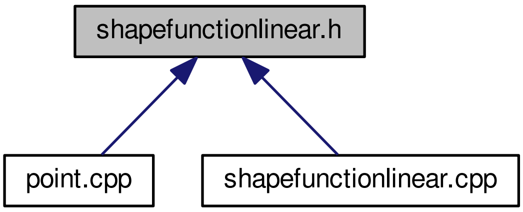
7.115 shapefunctionlinear.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionlinear.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ShapeFunctionLinear

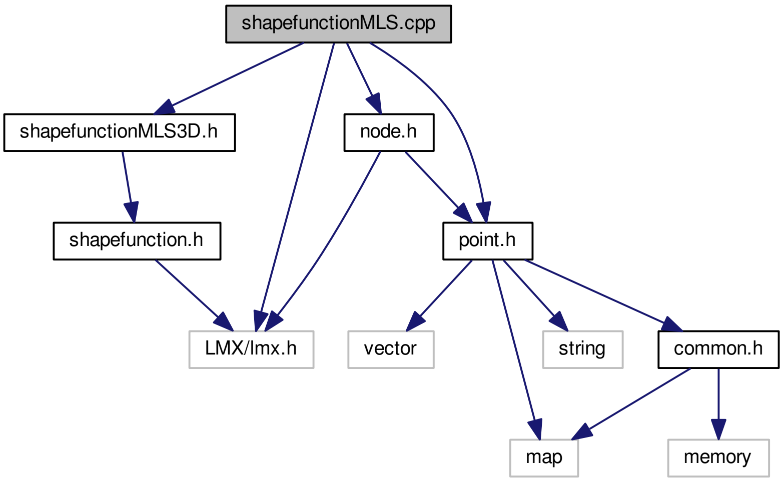
Namespaces

- 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

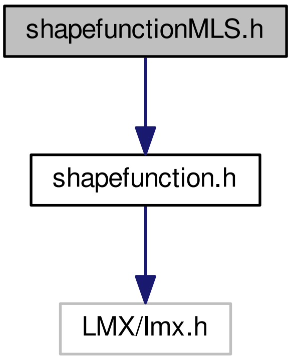
- mknix

7.117 shapefunctionMLS.h File Reference

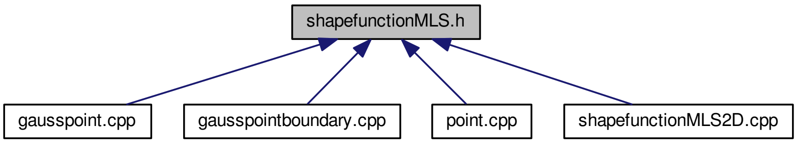
Function for aproximation of basic variables by Moving Least Squares fit.

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionMLS.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ShapeFunctionMLS

Namespaces

- mknix

7.117.1 Detailed Description

Function for aproximation of basic variables by Moving Least Squares fit.

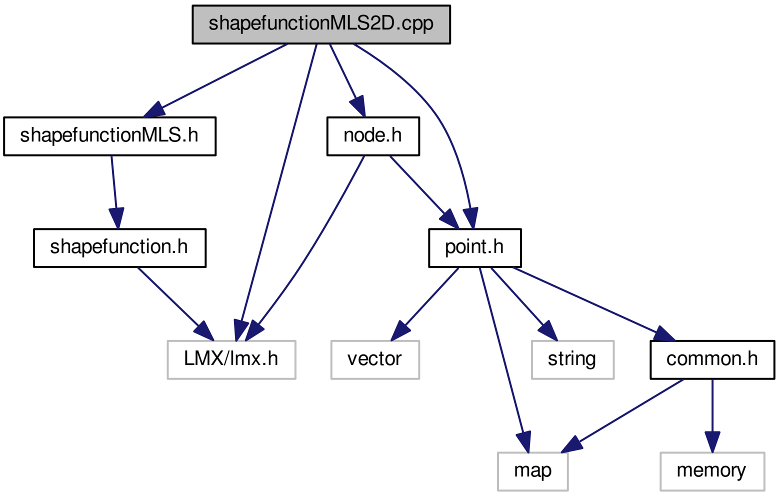
Author

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:



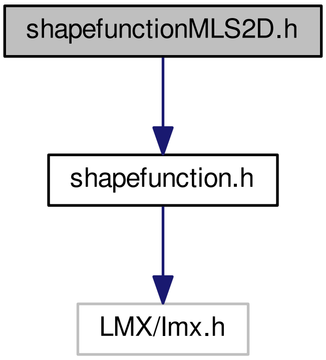
Namespaces

- mknix

7.119 shapefunctionMLS2D.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionMLS2D.h:



Classes

- class mknix::ShapeFunctionMLS

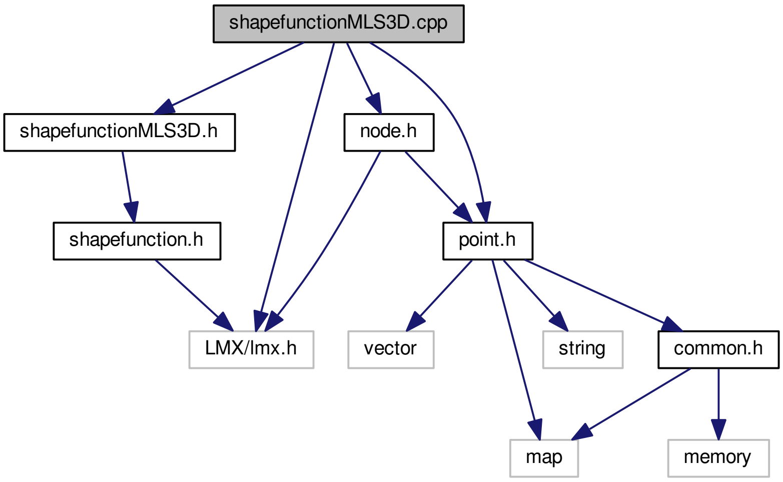
Namespaces

- 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:



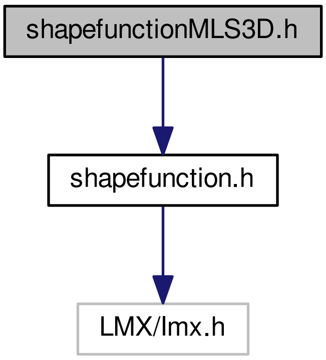
Namespaces

- mknix

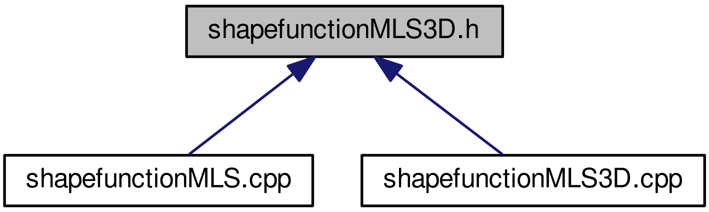
7.121 shapefunctionMLS3D.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctionMLS3D.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ShapeFunctionMLS

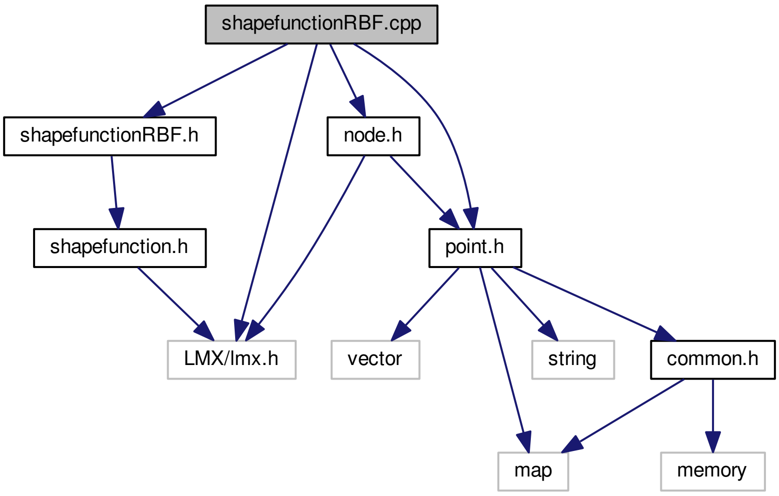
Namespaces

- mknix

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

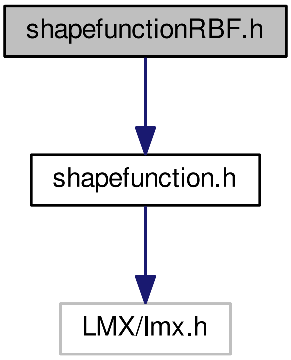
- mknix

7.123 shapefunctionRBF.h File Reference

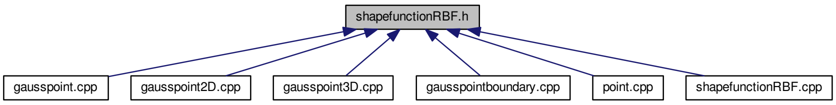
Function for interpolation of basic variables by means of Radial basis Functions.

```
#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.

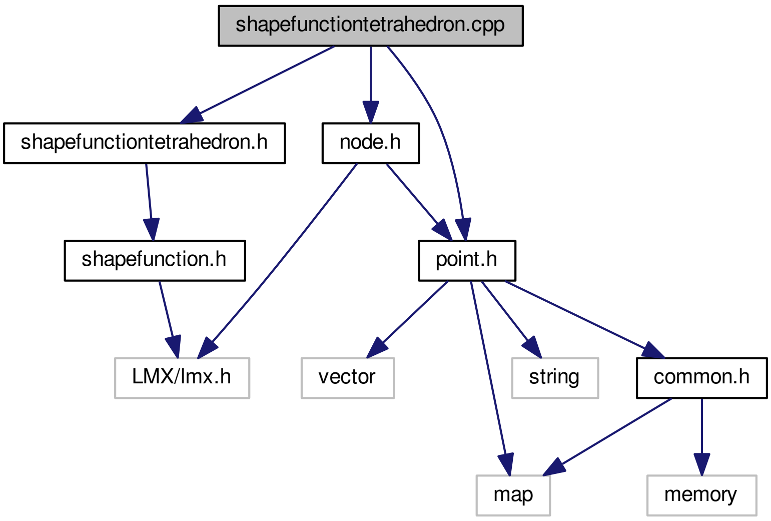
Author

Daniel Iglesias

7.124 shapefunctiontetrahedron.cpp File Reference

```
#include "shapefunctiontetrahedron.h"
#include "point.h"
#include "node.h"
```

Include dependency graph for shapefunctiontetrahedron.cpp:



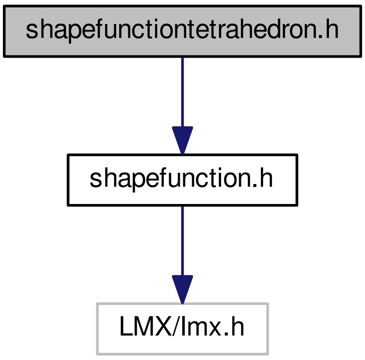
Namespaces

- mknix

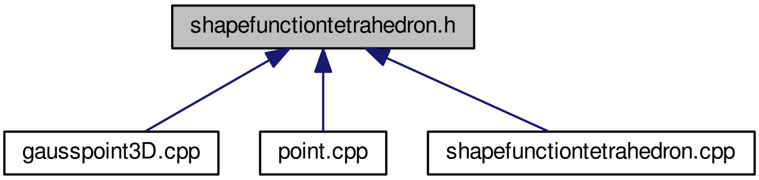
7.125 shapefunctiontetrahedron.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctiontetrahedron.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ShapeFunctionTetrahedron

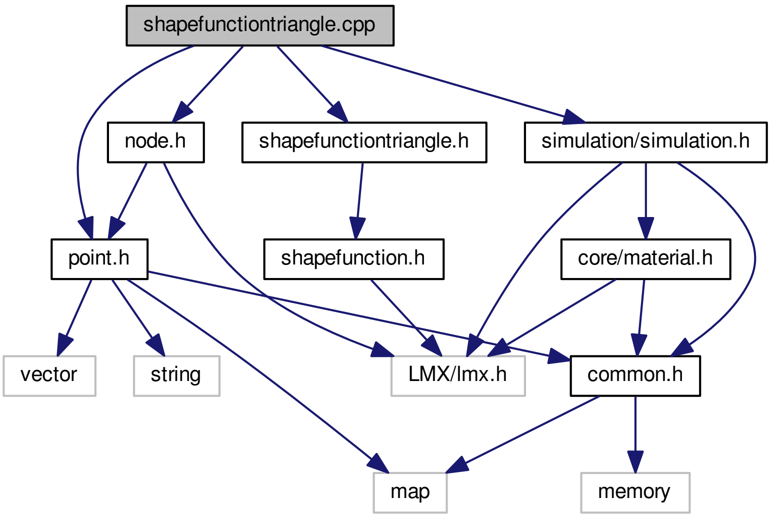
Namespaces

- 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:



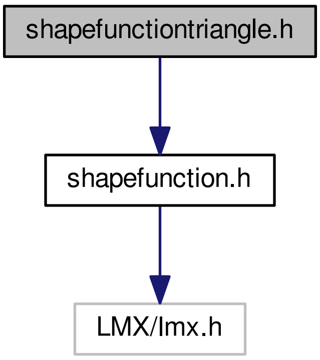
Namespaces

- mknix

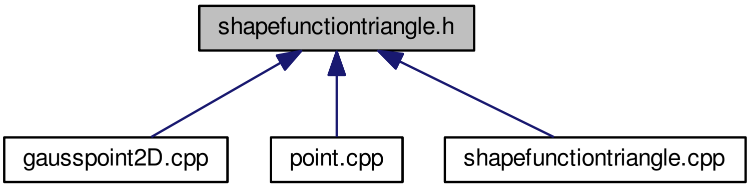
7.127 shapefunctiontriangle.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctiontriangle.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ShapeFunctionTriangle

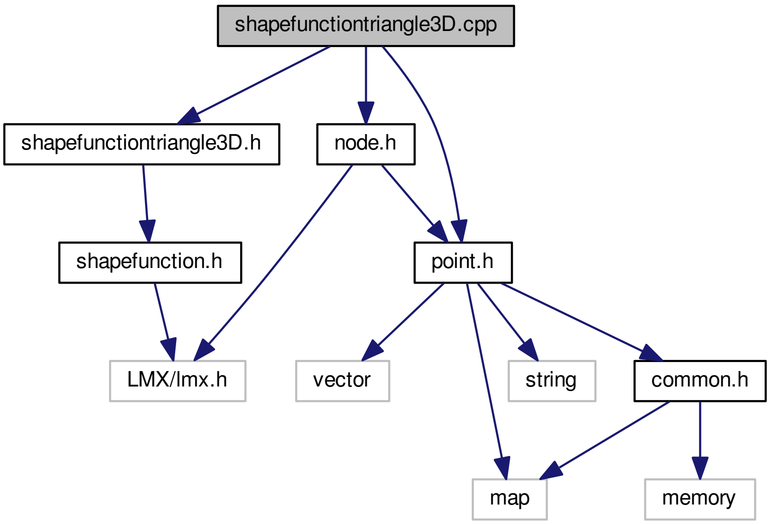
Namespaces

- mknix

7.128 shapefunctiontriangle3D.cpp File Reference

```
#include "shapefunctiontriangle3D.h"
#include "point.h"
#include "node.h"
```

Include dependency graph for shapefunctiontriangle3D.cpp:



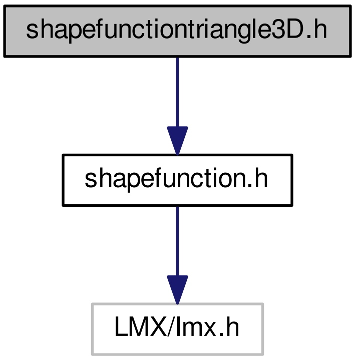
Namespaces

- mknix

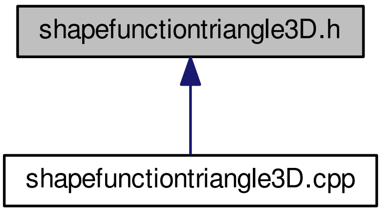
7.129 shapefunctiontriangle3D.h File Reference

```
#include "shapefunction.h"
```

Include dependency graph for shapefunctiontriangle3D.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::ShapeFunctionTriangleSigned

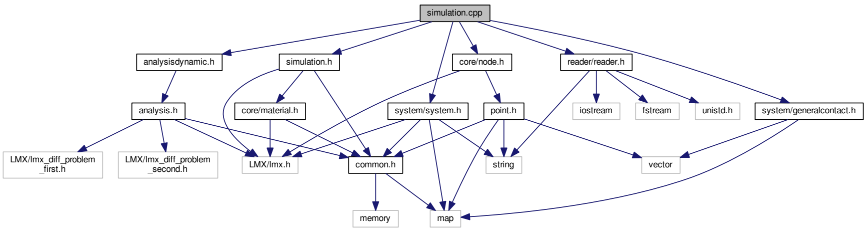
Namespaces

- 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

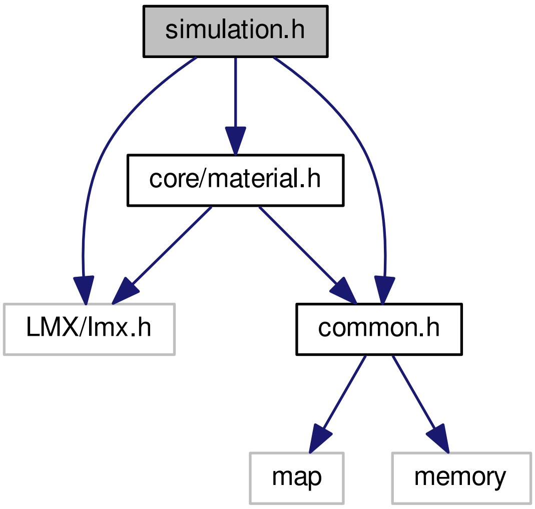
Functions

- 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

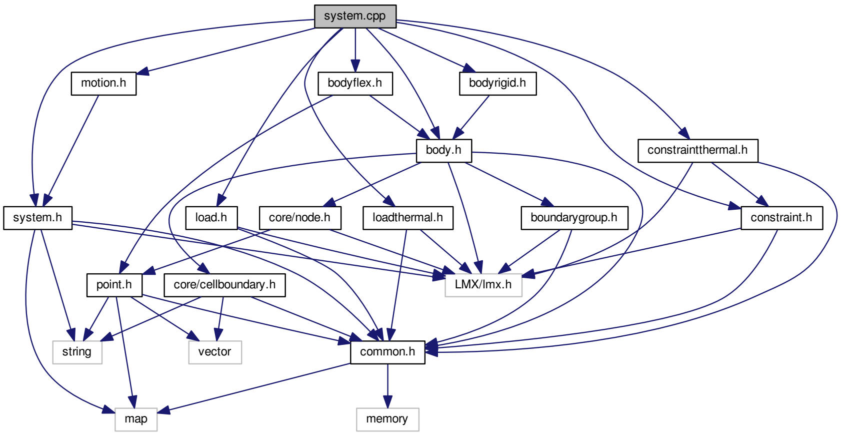
Namespaces

- 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:



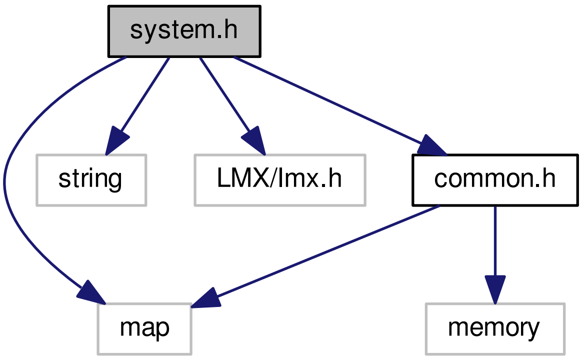
Namespaces

- mknix

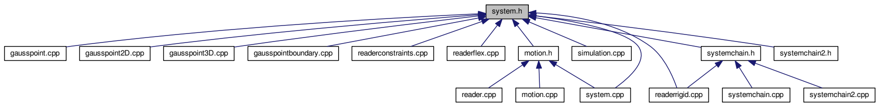
7.133 system.h File Reference

```
#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

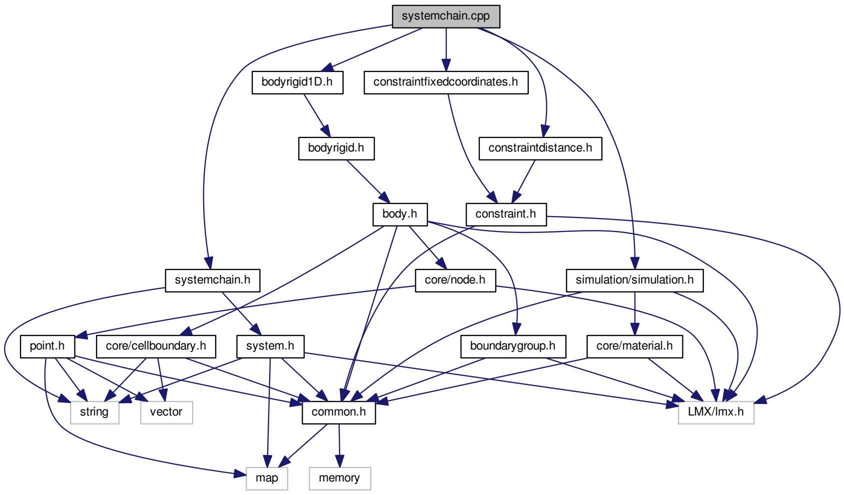
Namespaces

- 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:



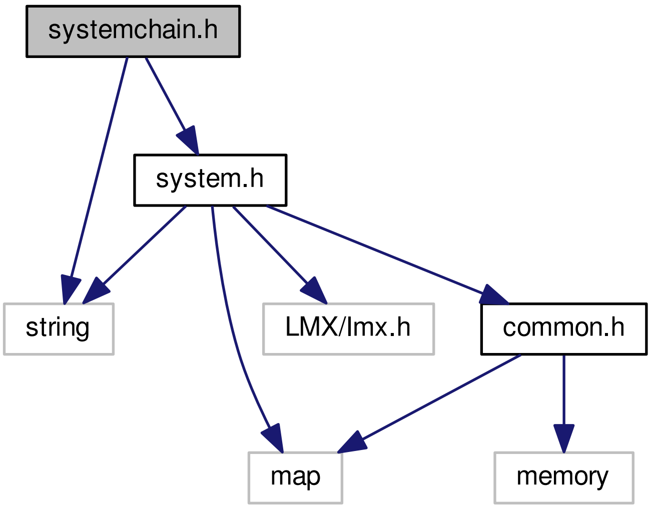
Namespaces

- mknix

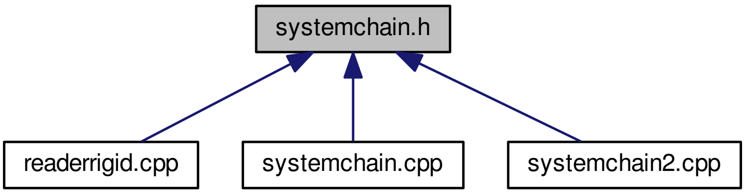
7.135 systemchain.h File Reference

```
#include "system.h"
#include <string>
```

Include dependency graph for systemchain.h:



This graph shows which files directly or indirectly include this file:



Classes

- class mknix::SystemChain

Namespaces

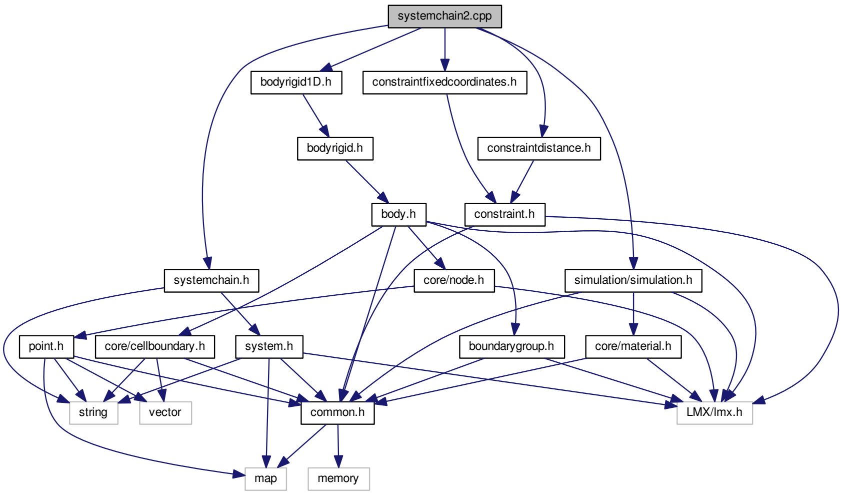
- mknix

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:



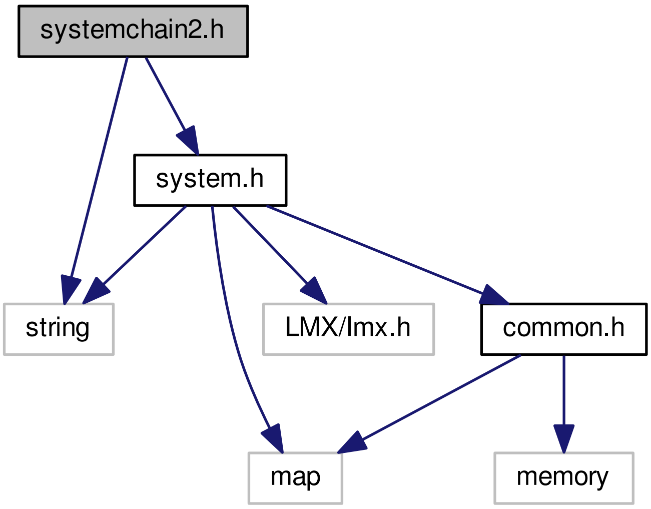
Namespaces

- 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