

mpm

Alpha

Generated by Doxygen 1.8.13

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Data Structure Index	5
3.1	Data Structures	5
4	Namespace Documentation	7
4.1	mfm Namespace Reference	7
4.1.1	Detailed Description	8
4.1.2	Typedef Documentation	8
4.1.2.1	Index	8
5	Data Structure Documentation	9
5.1	mfm::Blockset< Tdim > Class Template Reference	9
5.2	mfm::Domain< Tdim > Class Template Reference	9
5.2.1	Detailed Description	10
5.2.2	Constructor & Destructor Documentation	10
5.2.2.1	Domain()	10
5.2.3	Member Function Documentation	10
5.2.3.1	add_node()	11
5.2.3.2	assign_velocity_constraints()	11
5.2.3.3	create_nodes()	11

5.2.3.4	generate_material_points()	12
5.2.3.5	remove_node()	12
5.3	mfm::Element< dim > Class Template Reference	13
5.3.1	Detailed Description	13
5.4	mfm::IO Class Reference	13
5.4.1	Detailed Description	14
5.4.2	Constructor & Destructor Documentation	14
5.4.2.1	IO()	14
5.5	mfm::Material< dim > Class Template Reference	14
5.6	MaterialPoint< Tdim > Class Template Reference	14
5.6.1	Detailed Description	15
5.7	mfm::MFM Class Reference	15
5.7.1	Detailed Description	16
5.8	mfm::MFMEExplicit< Tdim > Class Template Reference	16
5.8.1	Detailed Description	16
5.8.2	Constructor & Destructor Documentation	17
5.8.2.1	MFMEExplicit()	17
5.9	mfm::Node< Tdim > Class Template Reference	17
5.9.1	Constructor & Destructor Documentation	17
5.9.1.1	Node()	17
5.9.2	Member Function Documentation	18
5.9.2.1	assign_coordinates()	18
5.10	mfm::Particle< dim > Class Template Reference	18
5.10.1	Detailed Description	18
5.11	mfm::ParticleBase< Tdim > Class Template Reference	19
5.11.1	Detailed Description	19
5.11.2	Constructor & Destructor Documentation	20
5.11.2.1	ParticleBase()	20
5.11.3	Member Function Documentation	20
5.11.3.1	assign_coordinates()	20
5.12	mfm::Quadrature< dim > Class Template Reference	20
5.13	mfm::ReadMesh< Tdim > Class Template Reference	21
5.13.1	Detailed Description	21
5.13.2	Constructor & Destructor Documentation	21
5.13.2.1	ReadMesh()	21
5.13.3	Member Function Documentation	22
5.13.3.1	read_mesh_cells()	22
5.13.3.2	read_nodes()	22

Chapter 1

Namespace Index

1.1 Namespace List

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

mfm	MFM namespace	7
---------------------	---	-------------------

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

mfm::Blockset< Tdim >	9
mfm::Domain< Tdim >	9
mfm::Element< dim >	13
mfm::IO	13
mfm::Material< dim >	14
MaterialPoint< Tdim >	14
mfm::MFM	15
mfm::MFMEexplicit< Tdim >	16
mfm::Node< Tdim >	17
mfm::ParticleBase< Tdim >	19
mfm::ParticleBase< dim >	19
mfm::Particle< dim >	18
mfm::Quadrature< dim >	20
mfm::ReadMesh< Tdim >	21

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

mfm::Blockset< Tdim >	9
mfm::Domain< Tdim >	
Class that stores information about the domain	9
mfm::Element< dim >	
Base class that stores the information about shape functions	13
mfm::IO	
Input/Output handler	13
mfm::Material< dim >	14
MaterialPoint< Tdim >	
Class that contains the material points of the meshfree problem	14
mfm::MFM	15
mfm::MFMEexplicit< Tdim >	
A class that implements the fully explicit mfm	16
mfm::Node< Tdim >	17
mfm::Particle< dim >	
Base class that stores the information about particles	18
mfm::ParticleBase< Tdim >	
Base class that stores the information about particleBases	19
mfm::Quadrature< dim >	20
mfm::ReadMesh< Tdim >	
Class that returns mesh and particles locataions based on GMesh ascii file	21

Chapter 4

Namespace Documentation

4.1 mfm Namespace Reference

MFM namespace.

Data Structures

- class [Blockset](#)
- class [Domain](#)
Class that stores information about the domain.
- class [Element](#)
Base class that stores the information about shape functions.
- class [IO](#)
Input/Output handler.
- class [Material](#)
- class [MFM](#)
- class [MFMEexplicit](#)
A class that implements the fully explicit mfm.
- class [Node](#)
- class [Particle](#)
Base class that stores the information about particles.
- class [ParticleBase](#)
Base class that stores the information about particleBases.
- class [Quadrature](#)
- class [ReadMesh](#)
Class that returns mesh and particles locataions based on GMesh ascii file.

Typedefs

- using [Index](#) = unsigned long long
Global index type for the node_base.

4.1.1 Detailed Description

[MFM](#) namespace.

4.1.2 Typedef Documentation

4.1.2.1 Index

```
typedef unsigned long long mfm::Index
```

Global index type for the node_base.

Global index type for the cell.

Global index type for the particleBase.

Chapter 5

Data Structure Documentation

5.1 mfm::Blockset< Tdim > Class Template Reference

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

- include/blockset.h

5.2 mfm::Domain< Tdim > Class Template Reference

Class that stores information about the domain.

```
#include <domain.h>
```

Public Types

- using [VectorDim](#) = Eigen::Matrix< double, Tdim, 1 >
Define a vector of size dimension.

Public Member Functions

- [Domain](#) (unsigned [id](#))
Constructor.
- [~Domain](#) ()=default
Default destructor.
- [Domain](#) (const [Domain](#)< Tdim > &)=delete
Delete copy constructor.
- [Domain](#) & [operator=](#) (const [Domain](#)< Tdim > &)=delete
Delete assignement operator.
- unsigned [id](#) () const
Return id of domain.
- bool [create_nodes](#) (mfm::Index gnid, const std::vector< [VectorDim](#) > &coordinates, bool check_↔
duplicates=true)

Create nodes from coordinates.

- bool [add_node](#) (const std::shared_ptr< [mfm::Node](#)< Tdim >> &node, bool check_duplicates=true)

Add a node to the domain.

- bool [remove_node](#) (const std::shared_ptr< [mfm::Node](#)< Tdim >> &node)
- [mfm::Index nnodes](#) () const

Number of nodes in the mesh.

- bool [assign_velocity_constraints](#) (const std::vector< std::tuple< [mfm::Index](#), unsigned, double >> &velocity_constraints)
- void [generate_material_points](#) (unsigned nquadratures=1)
- void [read_input_file](#) ()

Read mesh file.

5.2.1 Detailed Description

```
template<unsigned Tdim>
class mfm::Domain< Tdim >
```

Class that stores information about the domain.

[Domain](#) class

domain class which stores the particles, nodes, cells.

Template Parameters

<i>Tdim</i>	Dimension
-------------	-----------

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Domain()

```
template<unsigned Tdim>
mfm::Domain< Tdim >::Domain (
    unsigned id )
```

Constructor.

Parameters

in	<i>id</i>	Global mesh id
----	-----------	----------------

5.2.3 Member Function Documentation

5.2.3.1 add_node()

```
template<unsigned Tdim>
bool mfm::Domain< Tdim >::add_node (
    const std::shared_ptr< mfm::Node< Tdim >> & node,
    bool check_duplicates = true )
```

Add a node to the domain.

Add a node to the domain

Parameters

in	<i>node</i>	A shared pointer to node
in	<i>check_duplicates</i>	Parameter to check duplicates

Return values

<i>insertion_status</i>	Return the successful addition of a node
-------------------------	--

5.2.3.2 assign_velocity_constraints()

```
template<unsigned Tdim>
bool mfm::Domain< Tdim >::assign_velocity_constraints (
    const std::vector< std::tuple< mfm::Index, unsigned, double >> & velocity_←
constraints )
```

Assign velocity constraints to nodes

Parameters

in	<i>velocity_constraints</i>	Constraint at node, dir, and velocity
----	-----------------------------	---------------------------------------

5.2.3.3 create_nodes()

```
template<unsigned Tdim>
bool mfm::Domain< Tdim >::create_nodes (
    mfm::Index gnid,
    const std::vector< VectorDim > & coordinates,
    bool check_duplicates = true )
```

Create nodes from coordinates.

Create nodes from coordinates

Parameters

in	<i>gnid</i>	Global node id
in	<i>node_type</i>	Node type
in	<i>coordinates</i>	Nodal coordinates
in	<i>check_duplicates</i>	Parameter to check duplicates

Return values

<i>status</i>	Create node status
---------------	--------------------

5.2.3.4 generate_material_points()

```
template<unsigned Tdim>
void mfm::Domain< Tdim >::generate_material_points (
    unsigned nquadratures = 1 )
```

Generate points**Parameters**

in	<i>nquadratures</i>	Number of points per direction in cell
----	---------------------	--

Return values

<i>point</i>	Material point coordinates
--------------	--

5.2.3.5 remove_node()

```
template<unsigned Tdim>
bool mfm::Domain< Tdim >::remove_node (
    const std::shared_ptr< mfm::Node< Tdim >> & node )
```

Remove a node from the domain**Parameters**

in	<i>node</i>	A shared pointer to node
----	-------------	--------------------------

Return values

<i>insertion_status</i>	Return the successful addition of a node
-------------------------	--

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

- include/domain.h

5.3 mfm::Element< dim > Class Template Reference

Base class that stores the information about shape functions.

```
#include <element.h>
```

5.3.1 Detailed Description

```
template<unsigned dim>
class mfm::Element< dim >
```

Base class that stores the information about shape functions.

Base class of shape functions

Template Parameters

<i>Tdim</i>	Dimension
-------------	-----------

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

- include/element.h

5.4 mfm::IO Class Reference

Input/Output handler.

```
#include <io.h>
```

Public Member Functions

- [IO](#) (int argc, char **argv)
- unsigned [nthreads](#) () const
Return number of tbb threads.
- void [set_mesh_file_name](#) (std::string filename)
Sets the mesh file name.
- std::string [get_mesh_file_name](#) () const
Returns the mesh file name.

5.4.1 Detailed Description

Input/Output handler.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 IO()

```
mfm::IO::IO (
    int argc,
    char ** argv ) [inline]
```

Constructor with argc and argv

Parameters

in	<i>argc</i>	Number of input arguments
in	<i>argv</i>	Input arguments

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

- include/io.h

5.5 mfm::Material< dim > Class Template Reference

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

- include/particle_base.h

5.6 MaterialPoint< Tdim > Class Template Reference

Class that contains the material points of the meshfree problem.

```
#include <material_point.h>
```

Public Member Functions

- [MaterialPoint](#) ()
Default constructor.
- VectorDim **reference_coordinates** () const
- VectorDim **update_coordinates** ()

5.6.1 Detailed Description

```
template<unsigned Tdim>
class MaterialPoint< Tdim >
```

Class that contains the material points of the meshfree problem.

MaterialPoint

Template Parameters

<i>Tdim</i>	Dimension
-------------	-----------

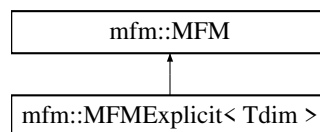
The documentation for this class was generated from the following file:

- include/material_point.h

5.7 mfm::MFM Class Reference

```
#include <mfm.h>
```

Inheritance diagram for mfm::MFM:



Public Member Functions

- **MFM** (std::unique_ptr< [IO](#) > &&io)
- virtual bool **initialise_mesh** ()=0
- virtual bool **initialise_particles** ()=0
- virtual bool **initialise_nodes** ()=0
Intialise nodes.
- virtual bool **solve** ()=0

Protected Attributes

- std::string **uuid_**
- double **dt_** {std::numeric_limits<double>::max()}
Time step size.
- bool **axi_** {false}
AXISYMMETRIC.
- **mfm::Index** **step_** {0}
Current step.
- **mfm::Index** **nsteps_** {std::numeric_limits<[mfm::Index](#)>::max()}
Number of steps.
- std::unique_ptr< [mfm::IO](#) > **io_**
A unique ptr to IO object.

5.7.1 Detailed Description

[MFM](#) class [MFM](#) class calls solver and algorithm mfm class: implicit and explicit mfm

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

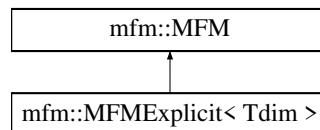
- include/mfm.h

5.8 mfm::MFMEExplicit< Tdim > Class Template Reference

A class that implements the fully explicit mfm.

```
#include <mfm_explicit.h>
```

Inheritance diagram for mfm::MFMEExplicit< Tdim >:



Public Member Functions

- [MFMEExplicit](#) (std::unique_ptr< [IO](#) > &&io)
Default constructor.
- bool [initialise_mesh](#) () override
Initialise mesh.
- bool [initialise_nodes](#) () override
Intialise nodes.
- bool [initialise_particles](#) () override
Initialise particles.
- bool [solve](#) () override
Solve.

Protected Attributes

- std::unique_ptr< [mfm::Domain](#)< Tdim > > [domain_](#)
[Domain](#) object.

5.8.1 Detailed Description

```
template<unsigned Tdim>
class mfm::MFMEExplicit< Tdim >
```

A class that implements the fully explicit mfm.

[MFMEExplicit](#) class

A single-phase explicit MPM

Template Parameters

<i>Tdim</i>	Dimension
-------------	-----------

5.8.2 Constructor & Destructor Documentation

5.8.2.1 MFMEexplicit()

```
template<unsigned Tdim>
mfm::MFMEexplicit< Tdim >::MFMEexplicit (
    std::unique_ptr< IO > && io )
```

Default constructor.

Constructor.

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

- include/mfm_explicit.h

5.9 mfm::Node< Tdim > Class Template Reference

Public Types

- using [VectorDim](#) = Eigen::Matrix< double, Tdim, 1 >
Define a vector of size dimension.

Public Member Functions

- [Node](#) (mfm::Index id, const [VectorDim](#) &coords)
- [~Node](#) ()
Destructor.
- [Node](#) (const [Node](#)< Tdim > &)=delete
Delete copy constructor.
- [Node](#) & [operator=](#) (const [Node](#)< Tdim > &)=delete
Delete assignement operator.
- void [assign_coordinates](#) (const [VectorDim](#) &coord)

5.9.1 Constructor & Destructor Documentation

5.9.1.1 Node()

```
template<unsigned Tdim>
mfm::Node< Tdim >::Node (
    mfm::Index id,
    const VectorDim & coords )
```

Constructor

Parameters

--	--

5.9.2 Member Function Documentation**5.9.2.1 assign_coordinates()**

```
template<unsigned Tdim>
void mfm::Node< Tdim >::assign_coordinates (
    const VectorDim & coord ) [inline]
```

Assign coordaintes

Parameters

in	coord	Assign coord as coordinates of the nodebase
----	-------	---

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

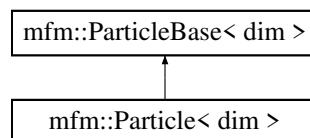
- include/node.h

5.10 mfm::Particle< dim > Class Template Reference

Base class that stores the information about particles.

```
#include <particle.h>
```

Inheritance diagram for mfm::Particle< dim >:

**Additional Inherited Members****5.10.1 Detailed Description**

```
template<unsigned dim>
class mfm::Particle< dim >
```

Base class that stores the information about particles.

[Particle](#) class

[Particle](#) class: id_ and coordinates.

Template Parameters

<i>Tdim</i>	Dimension
-------------	-----------

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

- include/particle.h

5.11 mfm::ParticleBase< Tdim > Class Template Reference

Base class that stores the information about particleBases.

```
#include <particle_base.h>
```

Public Types

- using [VectorDim](#) = Eigen::Matrix< double, Tdim, 1 >
Define a vector of size dimension.

Public Member Functions

- [ParticleBase](#) ([Index id](#), const [VectorDim](#) &coord)
- virtual [~ParticleBase](#) ()
Destructor.
- [Index id](#) () const
Return id of the particleBase.
- void [assign_coordinates](#) (const [VectorDim](#) &coord)

Protected Attributes

- [VectorDim](#) [coordinates_](#)
coordinates
- [Index](#) [id_](#)
particleBase iD
- int [cell_id](#)
Cell id.
- Eigen::Matrix< double, Tdim, 1 > [xi_](#)
Reference coordinates (in a cell)

5.11.1 Detailed Description

```
template<unsigned Tdim>
class mfm::ParticleBase< Tdim >
```

Base class that stores the information about particleBases.

[ParticleBase](#) class

[ParticleBase](#) class: [id_](#) and [coordinates](#).

Template Parameters

<i>Tdim</i>	Dimension
-------------	-----------

5.11.2 Constructor & Destructor Documentation

5.11.2.1 ParticleBase()

```
template<unsigned Tdim>
mfm::ParticleBase< Tdim >::ParticleBase (
    Index id,
    const VectorDim & coord )
```

Constructor with id and coordinates

Parameters

in	<i>id</i>	Particle id
in	<i>coord</i>	coordinates of the particle

5.11.3 Member Function Documentation

5.11.3.1 assign_coordinates()

```
template<unsigned Tdim>
void mfm::ParticleBase< Tdim >::assign_coordinates (
    const VectorDim & coord ) [inline]
```

Assign coordinates

Parameters

in	<i>coord</i>	Assign coord as coordinates of the particleBase
----	--------------	---

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

- include/particle_base.h

5.12 mfm::Quadrature< dim > Class Template Reference

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

- `include/quadrature.h`

5.13 mfm::ReadMesh< Tdim > Class Template Reference

Class that returns mesh and particles locataions based on GMesh ascii file.

```
#include <read_mesh.h>
```

Public Types

- using [VectorDim](#) = Eigen::Matrix< double, Tdim, 1 >
Define a vector of size dimension.

Public Member Functions

- [ReadMesh](#) ()
Constructor.
- [ReadMesh](#) (std::string filename)
- std::vector< std::vector< [mfm::Index](#) > > [read_mesh_cells](#) (const std::string &mesh)
- std::vector< [VectorDim](#) > [read_nodes](#) ()

5.13.1 Detailed Description

```
template<unsigned Tdim>
class mfm::ReadMesh< Tdim >
```

Class that returns mesh and particles locataions based on GMesh ascii file.

[ReadMesh](#) class

Template Parameters

<i>Tdim</i>	Dimension
-------------	-----------

5.13.2 Constructor & Destructor Documentation

5.13.2.1 ReadMesh()

```
template<unsigned Tdim>
mfm::ReadMesh< Tdim >::ReadMesh (
    std::string filename ) [inline]
```

Constructor with filename

Parameters

in	<i>filename</i>	with name of file
----	-----------------	-------------------

filename_

5.13.3 Member Function Documentation

5.13.3.1 read_mesh_cells()

```
template<unsigned Tdim>
std::vector<std::vector<mfm::Index> > mfm::ReadMesh< Tdim >::read_mesh_cells (
    const std::string & mesh )
```

Read mesh cells file

Parameters

in	<i>mesh</i>	file name with nodes and cells
----	-------------	--------------------------------

Return values

<i>cells</i>	Vector of nodal indices of cells
--------------	----------------------------------

5.13.3.2 read_nodes()

```
template<unsigned Tdim>
std::vector< Eigen::Matrix< double, Tdim, 1 > > mfm::ReadMesh< Tdim >::read_nodes ( )
```

Read particles file

Parameters

in	<i>particles_files</i>	file name with particle coordinates
----	------------------------	-------------------------------------

Return values

<i>coordinates</i>	Vector of particle coordinates
--------------------	--------------------------------

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

- include/read_mesh.h