Mesh2D

Library for interfacing with 2D unstructured and structured mesh

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 mesh_ns::Cell2D Class Reference

A cell in a 2D mesh. Contains all of the necessary functions access the geometrical elements of the cell.

`#include <cell2d.h>`

Public Member Functions

- Cell2D (std::vector< int > ids, Mesh2D *mesh)
  Constructor from global indexes and a mesh.
- std::vector< std::vector< int > > edges ()
  getter for the global indicies making up the edges of this cell
- std::vector< int > edge (int iL)
  getter for edge in cell
- std::vector< int > vertices ()
  getter for global indices of this cell
- std::vector< std::vector< double > > vertices_coords ()
  getter for returning the coordinates of the vertices
- int vertex (int iL)
  Converts from local cell index to global index.
- std::vector< double > vertex_coords (int iL)
  getter for the coordinates of a vertex
- std::vector< double > tangent (int iL)
  getter for vertex coords
- std::vector< Cell2D * > get_neighbours ()
  getter for the cell neighbours
- bool is_internal ()
  getter for internal condition
- bool is_boundary ()
  getter for boundary condition
- bool is_boundary_vertex (int iL)
  getter for vertex boundary condition
- bool is_boundary_edge (int iL)
  getter for edge boundary condition

- bool is_internal_vertex (int iL)
- double get_cell_area ()

  getter for cell area
- double get_cell_diam ()

  getter for cell diameter
- double get_edge_measure (int iL)

  Getter for the edge length.
- bool set_center (std::vector< double > center)

  Setter for barycentric coordinates.
- bool **add_neighbour** (Cell2D *cell)
- bool calc_cell_geometry_factors ()

  add a cell to the neighbour list
- std::vector< double > ari_coords ()

  calc cell diam, area etc
- std::vector< double > bary_coords ()

  arithmetic coordinates average of edge midpoints

2.1.1 Detailed Description

A cell in a 2D mesh. Contains all of the necessary functions access the geometrical elements of the cell.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 Cell2D()

```
Cell2D::Cell2D (
            std::vector< int > ids,
            Mesh2D * mesh )
```

Constructor from global indexes and a mesh.

Parameters

ids	vector containing the global indices of the cell vertices
mesh	pointer to the mesh that the cell is contained in

2.1.3 Member Function Documentation

2.1.3.1 edge()

```
std::vector< int > Cell2D::edge (
            int iL )
```

getter for edge in cell

Parameters

iL	local index of the edge going counter clockwise

Returns

container with the indices of the edges

2.1.3.2 edges()

```
std::vector< std::vector< int > > Cell2D::edges ( )
```

getter for the global indicies making up the edges of this cell

Returns

container of edges where the global indicies of the vertices making up the edges

2.1.3.3 get_cell_area()

```
double mesh_ns::Cell2D::get_cell_area ( )  [inline]
```

getter for cell area

Returns

2.1.3.4 get_cell_diam()

```
double mesh_ns::Cell2D::get_cell_diam ( )  [inline]
```

getter for cell diameter

Returns

2.1.3.5 get_edge_measure()

```
double Cell2D::get_edge_measure (
            int iL )
```

Getter for the edge length.

Parameters

iL	local coordinate of the edge

Returns

2.1.3.6 get_neighbours()

```
std::vector< Cell2D * > Cell2D::get_neighbours ( )
```

getter for the cell neighbours

Returns

vector containing the neighbouring cells

2.1.3.7 is_boundary()

```
bool mesh_ns::Cell2D::is_boundary ( )  [inline]
```

getter for boundary condition

Returns

true if boundary false if internal

2.1.3.8 is_boundary_edge()

```
bool Cell2D::is_boundary_edge (
            int iL )
```

getter for edge boundary condition

Parameters

iL	local coordinate of edge

Returns

true if boundary false if internal

2.1.3.9 is_boundary_vertex()

```
bool Cell2D::is_boundary_vertex (
            int iL )
```

getter for vertex boundary condition

Parameters

iL	local coordinate of vertex

Returns

true if boundary false if internal

2.1.3.10 is_internal()

```
bool mesh_ns::Cell2D::is_internal ( )  [inline]
```

getter for internal condition

Returns

true if internal false if boundary

2.1.3.11 is_internal_vertex()

```
bool Cell2D::is_internal_vertex (
            int iL )
```

Parameters

iL	

Returns

2.1.3.12 set_center()

```
bool mesh_ns::Cell2D::set_center (
            std::vector< double > center )  [inline]
```

Setter for barycentric coordinates.

Parameters

center	vector containing (x,y) coords of the cell barycenter

Returns

true if successfully added

2.1.3.13 tangent()

```
std::vector< double > Cell2D::tangent (
            int iL )
```

getter for vertex coords

Parameters

iL	local cell coordinate

Returns

container of vertex coordinates (x,y)

2.1.3.14 vertex()

```
int Cell2D::vertex (
            int iL )
```

Converts from local cell index to global index.

Parameters

iL	local index

Returns

global index

2.1.3.15 vertex_coords()

```
std::vector< double > Cell2D::vertex_coords (
            int iL )
```

getter for the coordinates of a vertex

Parameters

iL	local coordinate index

Returns

vector containing the x,y coordinates

2.1.3.16 vertices()

```
std::vector< int > Cell2D::vertices ( )  [inline]
```

getter for global indices of this cell

Returns

container with the indices of the cells

2.1.3.17 vertices_coords()

```
std::vector< std::vector< double > > Cell2D::vertices_coords ( )
```

getter for returning the coordinates of the vertices

Returns

a vector containing the vertices in the cell. They are ordered in counter clockwise order

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

- src/include/cell2d.h
- src/cell2d.cpp

2.2 mesh_ns::Mesh2D Class Reference

Class which represents a 2D mesh. It looks after the list of double precision vertices and a list of the cells in this mesh.

`#include <mesh2d.h>`

Public Member Functions

- Mesh2D ()
  default constructor for an empty mesh
- bool set_vertices_coords (std::vector< std::vector< double > > vertices_coords)
  Setter for the mesh vertices coordinates.
- bool **set_cell_edge** (std::vector< std::vector< size_t > >)
- bool **set_edge_vertex** (std::vector< std::vector< size_t > >)
- bool **set_edge_cell** (std::vector< std::vector< size_t > >)
- bool **set_vertex_edge** (std::vector< std::vector< size_t > >)
- bool **set_neighbours** (std::vector< std::vector< size_t > >)
- size_t **n_cells** ()
- size_t n_edges ()
  number of cells in the mesh
- size_t n_vertexs ()
  <
- size_t n_bcells ()
  <
- size_t n_bedges ()
  <
- size_t n_bvertexs ()
  <
- double xmin ()
  <
- double xmax ()
  <
- double ymin ()
  <
- double ymax ()
  <
- std::vector< double > get_vertex_coords (int i)
  <
- bool **add_cell** (Cell2D *cell)
- std::vector< Cell2D * > cells ()
  adds a cell to the mesh (this is only used by the mesh builder)

2.2.1 Detailed Description

Class which represents a 2D mesh. It looks after the list of double precision vertices and a list of the cells in this mesh.

2.2.2 Member Function Documentation

2.2.2.1 cells()

```
std::vector< Cell2D * > mesh_ns::Mesh2D::cells ( )  [inline]
```

adds a cell to the mesh (this is only used by the mesh builder)

<

2.2.2.2 get_vertex_coords()

```
std::vector< double > Mesh2D::get_vertex_coords (
            int i )
```

<

maximum y coordinate getter for the vertex coordinates

Parameters

i	global index of the vertex

Returns

vector containing coodinates

2.2.2.3 n_bcells()

```
size_t mesh_ns::Mesh2D::n_bcells ( )  [inline]
```

<

number of vertices in the mesh

2.2.2.4 n_bedges()

```
size_t mesh_ns::Mesh2D::n_bedges ( )  [inline]
```

<

number of boundary cells

2.2.2.5 n_bvertexs()

```
size_t mesh_ns::Mesh2D::n_bvertexs ( )  [inline]
```

<

number of boundary edges

2.2.2.6 n_vertexs()

```
size_t mesh_ns::Mesh2D::n_vertexs ( )  [inline]
```

<

number of edges in the mesh

2.2.2.7 set_vertices_coords()

```
bool mesh_ns::Mesh2D::set_vertices_coords (
            std::vector< std::vector< double > > vertices_coords )
```

Setter for the mesh vertices coordinates.

Parameters

vertices_coords	container of vertex coordinates ordered by the global ordering.

Returns

2.2.2.8 xmax()

```
double mesh_ns::Mesh2D::xmax ( )  [inline]
```

<

minimum x coordinate

2.2.2.9 xmin()

```
double mesh_ns::Mesh2D::xmin ( )  [inline]
```

<

number of boundary vertices

2.2.2.10 ymax()

```
double mesh_ns::Mesh2D::ymax ( )  [inline]
```

<

minimum y coordinate

2.2.2.11 ymin()

```
double mesh_ns::Mesh2D::ymin ( )  [inline]
```

<

maximum x coordinate

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

- src/include/mesh2d.h
- src/mesh2d.cpp

2.3 mesh_ns::Mesh2DBuilder Class Reference

Public Member Functions

- Mesh2DBuilder ()

  Constructor for Mesh2DBuilder.
- Mesh2D * build_the_mesh (std::vector< std::vector< double > > vertices, std::vector< std::vector< int > > cells)

  Build a Mesh2D from vertices and cells.
- Mesh2D * read_typ2_mesh (std::string file_name)

  use the typ2 file reader and build a Mesh2D object. This function just calls the Mesh2DImporter class and then build_the_mesh and is purerly here for convenience

2.3.1 Member Function Documentation

2.3.1.1 build_the_mesh()

```
Mesh2D * Mesh2DBuilder::build_the_mesh (
            std::vector< std::vector< double > > vertices,
            std::vector< std::vector< int > > cells )
```

Build a Mesh2D from vertices and cells.

Parameters

vertices	vector containing the coordinates of the vertices ordered using the global ordering. Note that indexes start at 0 in c++
cells	vector containing vectors with the global indexes of vertices making up a cell

Returns

a pointer to the mesh that was build

2.3.1.2 read_typ2_mesh()

```
Mesh2D* mesh_ns::Mesh2DBuilder::read_typ2_mesh (
            std::string file_name )
```

use the typ2 file reader and build a Mesh2D object. This function just calls the Mesh2DImporter class and then build_the_mesh and is purerly here for convenience

Parameters

file_name	name of the mesh file including path and extension!

Returns

Mesh2D object

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

- src/include/mesh2d_builder.h
- src/mesh2d_builder.cpp

2.4 mesh_ns::Mesh2dReaderTyp2 Class Reference

In charge of reading typ2 file and extracting the vertices coordinates, cell indices and the cell centers. Note that the global index is offset for c++ so a index of 1 becomes 0.

```
#include <import_mesh2d.h>
```

Public Member Functions

- Mesh2dReaderTyp2 (std::string file_name)

  Constructor for mesh reader.
- void read_mesh (std::vector< std::vector< double > > &vertices, std::vector< std::vector< int > > &cells, std::vector< std::vector< double > > &centers)

  read the file into the specified containers

2.4.1 Detailed Description

In charge of reading typ2 file and extracting the vertices coordinates, cell indices and the cell centers. Note that the global index is offset for c++ so a index of 1 becomes 0.

2.4.2 Constructor & Destructor Documentation

2.4.2.1 Mesh2dReaderTyp2()

```
Mesh2dReaderTyp2::Mesh2dReaderTyp2 (
            std::string file_name )
```

Constructor for mesh reader.

Parameters

file_name	name of the file name, needs to include the full path

2.4.3 Member Function Documentation

2.4.3.1 read_mesh()

```
void Mesh2dReaderTyp2::read_mesh (
            std::vector< std::vector< double > > & vertices,
            std::vector< std::vector< int > > & cells,
            std::vector< std::vector< double > > & centers )
```

read the file into the specified containers

Parameters

vertices	reference to a vector to hold the vertices coordinates
cells	reference to a vector to hold the cell indexes
centers	reference to a vector to hold the cell centers coordinates

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

- src/include/import_mesh2d.h
- src/import_mesh2d.cpp

2.5 mesh_ns::Mesh2DVtkWriter Class Reference

Public Member Functions

- **Mesh2DVtkWriter** (Mesh2D ∗mesh)
- bool **write_to_vtu** (std::string file_name)

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

- src/include/vtk_writer.h

Index