AFEPack for Electromagnetic

Generated by Doxygen 1.8.14

Contents

Chapter 1

Notes on AFEPack for Electromagnetic

In fact, I make it not only for Electromagnetic calculation, but also for the general PDE cases.

In this note, the variational principles for electromagnetic is mainly based on Jian-Ming Jin's the book “The Finite Element Method in Electromagnetics”.

1.1 Node Finite Elements

Consider the PED in a Lipschitz domain $\Omega \in \mathcal{R}^d$, $d = 2, 3$

$$-\nabla \cdot (c(x)\nabla u) + a(x)u = f \qquad \in \Omega$$

with right hand side $f \in L^2(\Omega)^d$.

The boundary condition is the homogeneous Dirichlet boundary condition

$$u = b \qquad on\ \partial\Omega_1$$

and the homogeneous Neumann boundary condition

$$\hat{n} \cdot c(x)\nabla u + q(x)u = g \qquad on\ \partial\Omega_2$$

provided that a,c,q is real or complex and $\hat{n}$is the outward normal from S, where $\partial\Omega_1 + \partial\Omega_2 = \partial\Omega$

1.1.1 Vartiational Formulation

by applying the first scalar Green's theorem

$$\int_\Omega v\nabla \cdot (c\nabla u) + c(\nabla v) \cdot (\nabla u)dV = \oint_{\partial\Omega} cv\frac{\partial u}{\partial n}dS$$

and the boundary conditions above, we can get

$$\int_\Omega c(\nabla v) \cdot (\nabla u)dV + \int_\Omega a(x)uvdV = \int_\Omega fvdV + \oint_{\partial\Omega_2} v(g - q(x)u)dS$$

the forms

$$a(u, v) := \int_\Omega c(\nabla v) \cdot (\nabla u)dV + \int_\Omega a(x)uvdV + \oint_{\partial\Omega_2} q(x)vudS$$

$$rhs(v) := \int_\Omega fvdV + \oint_{\partial\Omega_2} vgdS$$

1.1.2 Example1

The computational domain is $[-1,1]\times[-1,1]$. The parameters

$$c(x)=1,\quad a(x)=0\quad and\quad f=2\pi^2\sin(\pi x)\sin(\pi y)$$

the boundary parameters

$$b=0,q(x)=0,\quad and\quad g=\begin{cases}\pi\sin(\pi x) & x\in[-1,1]\quad and\quad y=-1\\ -\pi\sin(\pi y) & x=1\quad and\quad y\in[-1,1]\\ -\pi\sin(\pi x) & x\in[-1,1]\quad and\quad y=1\\ \pi\sin(\pi y) & x=-1\quad and\quad y\in[-1,1]\end{cases}$$

The exact solution is

$$u=\sin(\pi x)\sin(\pi y)$$

This figure shows the results of Example.

1.2 Vector Finite Elements

consider the vector-valued model problem

$$\nabla\times(c(x)\nabla\times\mathbf{u})+a(x)\mathbf{u}=\mathbf{f}\qquad\in\Omega$$

The boundary conditions are the homogenous Dirichlet boundary condition

$$\hat{n}\times\mathbf{u}=b\qquad on\ \partial\Omega_1$$

and the homogeneous Neumann boundary condition

$$c(x)\hat{n}\times(\nabla\times\mathbf{u})+q(x)\hat{n}\times(\hat{n}\times\mathbf{u})=\mathbf{g}\qquad on\ \partial\Omega_2$$

with $\partial\Omega_1+\partial\Omega_2=\partial\Omega$, a,c,and b,q are real(complex) numbers or functions.

1.2.1 Vartiational Formulation

Invoking the first vector Green's theorem and vector identity

$$\int_\Omega(c\nabla\times\mathbf{v})\cdot(\nabla\times\mathbf{u})-\mathbf{v}\cdot(\nabla\times(c\nabla\times\mathbf{u}))dV=\oint_{\partial\Omega}c(\mathbf{v}\times\nabla\times\mathbf{u})\cdot\hat{n}dS$$

$$(\mathbf{v}\times\nabla\times\mathbf{u})\cdot\hat{n}=(\hat{n}\times\mathbf{v})\cdot(\nabla\times\mathbf{u})=-\mathbf{v}\cdot(\hat{n}\times(\nabla\times\mathbf{u}))$$

applying the boundary conditions we obtain

$$\int_\Omega c(\nabla\times\mathbf{v})\cdot(\nabla\times\mathbf{u})dV+\int_\Omega a(x)\mathbf{u}\cdot\mathbf{v}dV=\int_\Omega\mathbf{f}\cdot\mathbf{v}dV-\oint_{\partial\Omega}\mathbf{v}\cdot(\mathbf{g}-q(x)\hat{n}\times(\hat{n}\times\mathbf{u}))dS$$

the forms

$$a(\mathbf{u},\mathbf{v}):=\int_\Omega c(\nabla\times\mathbf{v})\cdot(\nabla\times\mathbf{u})dV+\int_\Omega a(x)\mathbf{u}\cdot\mathbf{v}dV+\oint_{\partial\Omega_2}q(x)(\hat{n}\times\mathbf{u})\cdot(\hat{n}\times\mathbf{v})dS$$

$$rhs(\mathbf{v}):=\int_\Omega\mathbf{f}\cdot\mathbf{v}dV-\oint_{\partial\Omega}\mathbf{v}\cdot\mathbf{g}dS$$

1.2.2 Example

The computational domain is $[-1,1] \times [-1,1]$. The parameters

$$c(x) = 1, \quad a(x) = 1 \quad and \quad f = \begin{pmatrix} (2\pi^2+1)\cos(\pi x)\sin(\pi y) \\ -(2\pi^2+1)\sin(\pi x)\cos(\pi y) \end{pmatrix}$$

the boundary parameters

$$b = 0, q(x) = 0, \quad and \quad g = \begin{cases} (-2\pi\cos(\pi x), 0) & x \in [-1,1] \quad and \quad y = -1 \\ (0, -2\pi\cos(\pi y)) & x = 1 \quad and \quad y \in [-1,1] \\ (2\pi\cos(\pi x), 0) & x \in [-1,1] \quad and \quad y = 1 \\ (0, 2\pi\cos(\pi y)) & x = -1 \quad and \quad y \in [-1,1] \end{cases}$$

The exact solution is

$$u = \begin{pmatrix} \cos(\pi x)\sin(\pi y) \\ -\sin(\pi x)\cos(\pi y) \end{pmatrix}$$

This figure shows the results of Example

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

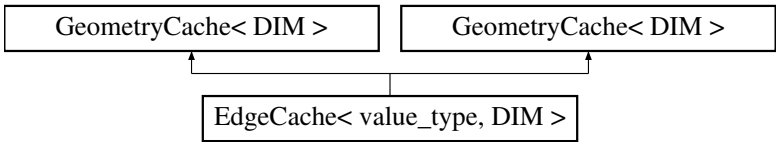
Chapter 5

Class Documentation

5.1 EdgeCache< value_type, DIM > Struct Template Reference

`#include <datacache.h>`

Inheritance diagram for EdgeCache< value_type, DIM >:



Public Attributes

- std::vector< std::vector< value_type > > basis_value
- std::vector< std::vector< std::vector< value_type > > > basis_gradient
- std::vector< std::vector< value_type > > un

5.1.1 Member Data Documentation

5.1.1.1 basis_gradient

```
template<typename value_type , int DIM>
std::vector< std::vector< std::vector< value_type > > > EdgeCache< value_type, DIM >↩
::basis_gradient
```

5.1.1.2 basis_value

```
template<typename value_type , int DIM>
std::vector< std::vector< value_type > > EdgeCache< value_type, DIM >::basis_value
```

5.1.1.3 un

```
template<typename value_type , int DIM>
std::vector< std::vector< value_type > > EdgeCache< value_type, DIM >::un
```

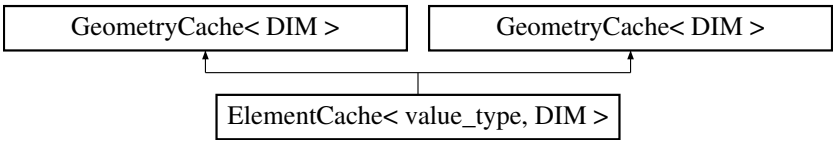
The documentation for this struct was generated from the following file:

- complex_edge_THFEM/datacache.h

5.2 ElementCache< value_type, DIM > Struct Template Reference

```
#include <datacache.h>
```

Inheritance diagram for ElementCache< value_type, DIM >:



Public Attributes

- double ind
- std::vector< std::vector< value_type > > basis_value

  index of the element
- std::vector< std::vector< std::vector< value_type > > > basis_gradient

5.2.1 Member Data Documentation

5.2.1.1 basis_gradient

```
template<typename value_type , int DIM>
std::vector< std::vector< std::vector< value_type > > > ElementCache< value_type, DIM >↩
::basis_gradient
```

5.2.1.2 basis_value

```
template<typename value_type , int DIM>
std::vector< std::vector< value_type > > ElementCache< value_type, DIM >::basis_value
```

index of the element

5.2.1.3 ind

```
template<typename value_type , int DIM>
double ElementCache< value_type, DIM >::ind
```

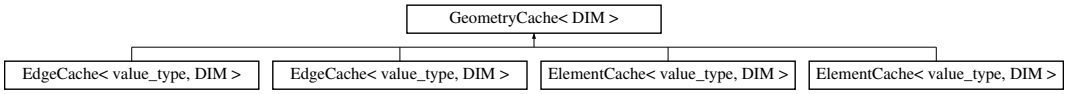
The documentation for this struct was generated from the following file:

- complex_edge_THFEM/datacache.h

5.3 GeometryCache< DIM > Struct Template Reference

```
#include <datacache.h>
```

Inheritance diagram for GeometryCache< DIM >:



Public Attributes

- Point< DIM > bc
- int n_quad_pnt
- std::vector< Point< DIM > > q_pnt
- std::vector< double > Jxw

5.3.1 Member Data Documentation

5.3.1.1 bc

```
template<int DIM>
Point< DIM > GeometryCache< DIM >::bc
```

5.3.1.2 Jxw

```
template<int DIM>
std::vector< double > GeometryCache< DIM >::Jxw
```

5.3.1.3 n_quad_pnt

```
template<int DIM>
int GeometryCache< DIM >::n_quad_pnt
```

5.3.1.4 q_pnt

```
template<int DIM>
std::vector< Point< DIM > > GeometryCache< DIM >::q_pnt
```

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

- complex_edge_THFEM/datacache.h

5.4 SolutionCache< value_type > Struct Template Reference

```
#include <datacache.h>
```

Public Attributes

- std::vector< value_type > val

5.4.1 Member Data Documentation

5.4.1.1 val

```
template<class value_type >
std::vector< value_type > SolutionCache< value_type >::val
```

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

- complex_edge_THFEM/datacache.h

5.5 uiExperiment Class Reference

`#include <uiexp.h>`

Public Member Functions

- uiExperiment (const std::string &file)

  time step
- virtual ~uiExperiment ()
- void init ()
- void buildFEMSpace ()
- void buildDGFEMSpace (int bmark=2)
- void updateGeometryCache (u_int alg_acc=3)
- void updateDGGeometryCache (u_int alg_acc=3)
- void DirichletBC (CFunc bnd, int bmark=1)
- void NeummanBC (CvFunc g, int bmark=2)
- void getRhs ()
- void getMat ()
- void solve ()
- void getError ()
- void adaptMesh ()
- void getIndicator ()
- virtual void saveData ()
- void run ()
- uiExperiment (const std::string &file)

  time step
- virtual ~uiExperiment ()
- void init ()
- void buildFEMSpace ()
- void buildDGFEMSpace (int bmark=2)
- void updateGeometryCache (u_int alg_acc=3)
- void updateDGGeometryCache (u_int alg_acc=3)
- void DirichletBC (CFunc bnd, int bmark=1)
- void NeummanBC (CFunc g, int bmark=2)
- void getRhs ()
- void getMat ()
- void solve ()
- void getError ()
- void adaptMesh ()
- void getIndicator ()
- virtual void saveData ()
- void run ()

Private Attributes

- HGeometryTree< DIM > h_tree
- IrregularMesh< DIM > ir_mesh
- std::string mesh_file
- TemplateGeometry< DIM > triangle_template_geometry

  mesh file
- CoordTransform< DIM, DIM > triangle_coord_transform
- TemplateDOF< DIM > triangle_template_dof
- BasisFunctionAdmin< vec_type, DIM, DIM > triangle_basis_function
- UnitOutNormal< DIM > triangle_unit_out_normal
- TemplateGeometry< DIM > twin_triangle_template_geometry
- CoordTransform< DIM, DIM > twin_triangle_coord_transform
- TemplateDOF< DIM > twin_triangle_template_dof
- BasisFunctionAdmin< vec_type, DIM, DIM > twin_triangle_basis_function
- UnitOutNormal< DIM > twin_triangle_unit_out_normal
- TemplateGeometry< DIM-1 > interval_template_geometry
- CoordTransform< DIM-1, DIM > interval_to2d_coord_transform
- std::vector< TemplateElement< vec_type, DIM, DIM > > template_element
- std::vector< TemplateDGElement< DIM-1, DIM > > dg_template_element
- DGFEMSpace< vec_type, DIM > fem_space
- std::vector< ElementCache< vec_type, DIM > > element_cache

  finite element space
- std::vector< EdgeCache< double, DIM > > edge_cache
- FEMFunction< vec_type, DIM > u_re
- FEMFunction< vec_type, DIM > u_im
- Eigen::SparseMatrix< cvaltype, Eigen::RowMajor > stiff_matrix
- Eigen::Matrix< cvaltype, Eigen::Dynamic, 1 > solution
- Eigen::Matrix< cvaltype, Eigen::Dynamic, 1 > rhs
- std::vector< T > triplets
- Indicator< DIM > indicator
- MeshAdaptor< DIM > mesh_adaptor
- double sys_t0 = 0
- double dt = 1

  system start time
- BasisFunctionAdmin< double, DIM, DIM > triangle_basis_function
- BasisFunctionAdmin< double, DIM, DIM > twin_triangle_basis_function
- std::vector< TemplateElement< double, DIM, DIM > > template_element
- DGFEMSpace< double, DIM > fem_space
- std::vector< ElementCache< double, DIM > > element_cache

  finite element space
- FEMFunction< double, DIM > u_re
- FEMFunction< double, DIM > u_im

5.5.1 Constructor & Destructor Documentation

5.5.1.1 uiExperiment() [1/2]

```
uiExperiment::uiExperiment (
            const std::string & file )
```

time step

5.5.1.2 ~uiExperiment() [1/2]

```
uiExperiment::~uiExperiment ( )  [virtual]
```

5.5.1.3 uiExperiment() [2/2]

```
uiExperiment::uiExperiment (
            const std::string & file )
```

time step

5.5.1.4 ~uiExperiment() [2/2]

```
virtual uiExperiment::~uiExperiment ( )  [virtual]
```

5.5.2 Member Function Documentation

5.5.2.1 adaptMesh() [1/2]

```
void uiExperiment::adaptMesh ( )
```

5.5.2.2 adaptMesh() [2/2]

```
void uiExperiment::adaptMesh ( )
```

5.5.2.3 buildDGFEMSpace() [1/2]

```
void uiExperiment::buildDGFEMSpace (
            int bmark = 2 )
```

5.5.2.4 buildDGFEMSpace() [2/2]

```
void uiExperiment::buildDGFEMSpace (
            int bmark = 2 )
```

5.5.2.5 buildFEMSpace() [1/2]

```
void uiExperiment::buildFEMSpace ( )
```

5.5.2.6 buildFEMSpace() [2/2]

```
void uiExperiment::buildFEMSpace ( )
```

5.5.2.7 DirichletBC() [1/2]

```
void uiExperiment::DirichletBC (
            CFunc bnd,
            int bmark = 1 )
```

5.5.2.8 DirichletBC() [2/2]

```
void uiExperiment::DirichletBC (
            CFunc bnd,
            int bmark = 1 )
```

5.5.2.9 getError() [1/2]

```
void uiExperiment::getError ( )
```

5.5.2.10 getError() [2/2]

```
void uiExperiment::getError ( )
```

5.5.2.11 getIndicator() [1/2]

```
void uiExperiment::getIndicator ( )
```

5.5.2.12 getIndicator() [2/2]

```
void uiExperiment::getIndicator ( )
```

5.5.2.13 getMat() [1/2]

```
void uiExperiment::getMat ( )
```

5.5.2.14 getMat() [2/2]

```
void uiExperiment::getMat ( )
```

5.5.2.15 getRhs() [1/2]

```
void uiExperiment::getRhs ( )
```

5.5.2.16 getRhs() [2/2]

```
void uiExperiment::getRhs ( )
```

5.5.2.17 init() [1/2]

```
void uiExperiment::init ( )
```

5.5.2.18 init() [2/2]

```
void uiExperiment::init ( )
```

5.5.2.19 NeummanBC() [1/2]

```
void uiExperiment::NeummanBC (
            CvFunc g,
            int bmark = 2 )
```

5.5.2.20 NeummanBC() [2/2]

```
void uiExperiment::NeummanBC (
            CFunc g,
            int bmark = 2 )
```

5.5.2.21 run() [1/2]

```
void uiExperiment::run ( )
```

5.5.2.22 run() [2/2]

```
void uiExperiment::run ( )
```

5.5.2.23 saveData() [1/2]

```
void uiExperiment::saveData ( )  [virtual]
```

5.5.2.24 saveData() [2/2]

```
virtual void uiExperiment::saveData ( )  [virtual]
```

5.5.2.25 solve() [1/2]

```
void uiExperiment::solve ( )
```

5.5.2.26 solve() [2/2]

```
void uiExperiment::solve ( )
```

5.5.2.27 updateDGGeometryCache() [1/2]

```
void uiExperiment::updateDGGeometryCache (
            u_int alg_acc = 3 )
```

5.5.2.28 updateDGGeometryCache() [2/2]

```
void uiExperiment::updateDGGeometryCache (
            u_int alg_acc = 3 )
```

5.5.2.29 updateGeometryCache() [1/2]

```
void uiExperiment::updateGeometryCache (
            u_int alg_acc = 3 )
```

5.5.2.30 updateGeometryCache() [2/2]

```
void uiExperiment::updateGeometryCache (
            u_int alg_acc = 3 )
```

5.5.3 Member Data Documentation

5.5.3.1 dg_template_element

```
std::vector< TemplateDGElement< DIM-1, DIM > > uiExperiment::dg_template_element  [private]
```

5.5.3.2 dt

```
double uiExperiment::dt = 1  [private]
```

system start time

5.5.3.3 edge_cache

```
std::vector< EdgeCache< double, DIM > > uiExperiment::edge_cache  [private]
```

5.5.3.4 element_cache [1/2]

```
std::vector<ElementCache<double,DIM> > uiExperiment::element_cache  [private]
```

finite element space

5.5.3.5 element_cache [2/2]

```
std::vector<ElementCache<vec_type,DIM> > uiExperiment::element_cache  [private]
```

finite element space

5.5.3.6 fem_space [1/2]

```
DGFEMSpace<double,DIM> uiExperiment::fem_space  [private]
```

5.5.3.7 fem_space [2/2]

```
DGFEMSpace<vec_type,DIM> uiExperiment::fem_space  [private]
```

5.5.3.8 h_tree

```
HGeometryTree< DIM > uiExperiment::h_tree  [private]
```

5.5.3.9 indicator

```
Indicator< DIM > uiExperiment::indicator  [private]
```

5.5.3.10 interval_template_geometry

```
TemplateGeometry< DIM-1 > uiExperiment::interval_template_geometry  [private]
```

5.5.3.11 interval_to2d_coord_transform

```
CoordTransform< DIM-1, DIM > uiExperiment::interval_to2d_coord_transform  [private]
```

5.5.3.12 ir_mesh

```
IrregularMesh< DIM > uiExperiment::ir_mesh  [private]
```

5.5.3.13 mesh_adaptor

```
MeshAdaptor< DIM > uiExperiment::mesh_adaptor  [private]
```

5.5.3.14 mesh_file

```
std::string uiExperiment::mesh_file  [private]
```

5.5.3.15 rhs

```
Eigen::Matrix< cvaltype, Eigen::Dynamic, 1 > uiExperiment::rhs  [private]
```

5.5.3.16 solution

```
Eigen::Matrix< cvaltype, Eigen::Dynamic, 1 > uiExperiment::solution  [private]
```

5.5.3.17 stiff_matrix

```
Eigen::SparseMatrix< cvaltype, Eigen::RowMajor > uiExperiment::stiff_matrix  [private]
```

5.5.3.18 sys_t0

```
double uiExperiment::sys_t0 = 0  [private]
```

5.5.3.19 template_element [1/2]

```
std::vector<TemplateElement<double,DIM,DIM> > uiExperiment::template_element  [private]
```

5.5.3.20 template_element [2/2]

```
std::vector<TemplateElement<vec_type,DIM,DIM> > uiExperiment::template_element  [private]
```

5.5.3.21 triangle_basis_function [1/2]

```
BasisFunctionAdmin<double,DIM,DIM> uiExperiment::triangle_basis_function  [private]
```

5.5.3.22 triangle_basis_function [2/2]

```
BasisFunctionAdmin<vec_type,DIM,DIM> uiExperiment::triangle_basis_function  [private]
```

5.5.3.23 triangle_coord_transform

```
CoordTransform< DIM, DIM > uiExperiment::triangle_coord_transform  [private]
```

5.5.3.24 triangle_template_dof

```
TemplateDOF< DIM > uiExperiment::triangle_template_dof  [private]
```

5.5.3.25 triangle_template_geometry

```
TemplateGeometry< DIM > uiExperiment::triangle_template_geometry  [private]
```

mesh file

5.5.3.26 triangle_unit_out_normal

```
UnitOutNormal< DIM > uiExperiment::triangle_unit_out_normal  [private]
```

5.5.3.27 triplets

```
std::vector< T > uiExperiment::triplets  [private]
```

5.5.3.28 twin_triangle_basis_function [1/2]

```
BasisFunctionAdmin<double,DIM,DIM> uiExperiment::twin_triangle_basis_function  [private]
```

5.5.3.29 twin_triangle_basis_function [2/2]

```
BasisFunctionAdmin<vec_type,DIM,DIM> uiExperiment::twin_triangle_basis_function  [private]
```

5.5.3.30 twin_triangle_coord_transform

```
CoordTransform< DIM, DIM > uiExperiment::twin_triangle_coord_transform  [private]
```

5.5.3.31 twin_triangle_template_dof

```
TemplateDOF< DIM > uiExperiment::twin_triangle_template_dof  [private]
```

5.5.3.32 twin_triangle_template_geometry

```
TemplateGeometry< DIM > uiExperiment::twin_triangle_template_geometry  [private]
```

5.5.3.33 twin_triangle_unit_out_normal

```
UnitOutNormal< DIM > uiExperiment::twin_triangle_unit_out_normal  [private]
```

5.5.3.34 u_im [1/2]

```
FEMFunction<double,DIM> uiExperiment::u_im  [private]
```

5.5.3.35 u_im [2/2]

```
FEMFunction<vec_type,DIM> uiExperiment::u_im  [private]
```

5.5.3.36 u_re [1/2]

```
FEMFunction<double,DIM> uiExperiment::u_re  [private]
```

5.5.3.37 u_re [2/2]

```
FEMFunction<vec_type,DIM> uiExperiment::u_re  [private]
```

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

- complex_edge_THFEM/uiexp.h
- complex_edge_THFEM/datacache.cpp
- complex_edge_THFEM/uiexp.cpp

Chapter 6

File Documentation

6.1 complex_edge_THFEM/D.d File Reference

6.2 complex_node_THFEM/D.d File Reference

6.3 complex_edge_THFEM/datacache.cpp File Reference

```
#include "uiexp.h"
#include "datacache.h"
```

Functions

- template<class value_type , int DIM, int DOW = DIM, int TDIM = DIM>
  void updateElementGeometryInfo (Element< value_type, DIM > &ele, u_int alg_acc, ElementCache< value_type, DIM > &ec)

6.3.1 Function Documentation

6.3.1.1 updateElementGeometryInfo()

```
template<class value_type , int DIM, int DOW = DIM, int TDIM = DIM>
void updateElementGeometryInfo (
            Element< value_type, DIM > & ele,
            u_int alg_acc,
            ElementCache< value_type, DIM > & ec )
```

6.4 complex_node_THFEM/datacache.cpp File Reference

```
#include "uiexp.h"
#include "datacache.h"
```

Functions

- template<class value_type , int DIM, int DOW = DIM, int TDIM = DIM>
  void updateElementGeometryInfo (Element< value_type, DIM > &ele, u_int alg_acc, ElementCache< value_type, DIM > &ec)
- template<class value_type , int DIM, int DOW = DIM, int TDIM = DIM>
  void updateEdgeGeometryInfo (DGElement< value_type, DIM > &edge, u_int alg_acc, EdgeCache< value_type, DIM > &ec)

6.4.1 Function Documentation

6.4.1.1 updateEdgeGeometryInfo()

```
template<class value_type , int DIM, int DOW = DIM, int TDIM = DIM>
void updateEdgeGeometryInfo (
            DGElement< value_type, DIM > & edge,
            u_int alg_acc,
            EdgeCache< value_type, DIM > & ec )
```

6.4.1.2 updateElementGeometryInfo()

```
template<class value_type , int DIM, int DOW = DIM, int TDIM = DIM>
void updateElementGeometryInfo (
            Element< value_type, DIM > & ele,
            u_int alg_acc,
            ElementCache< value_type, DIM > & ec )
```

6.5 complex_edge_THFEM/datacache.h File Reference

```
#include <lac/full_matrix.h>
```

Classes

- struct SolutionCache< value_type >
- struct GeometryCache< DIM >
- struct EdgeCache< value_type, DIM >
- struct ElementCache< value_type, DIM >

6.6 complex_node_THFEM/datacache.h File Reference

```
#include <lac/full_matrix.h>
```

Classes

- struct SolutionCache< value_type >
- struct GeometryCache< DIM >
- struct EdgeCache< value_type, DIM >
- struct ElementCache< value_type, DIM >

6.7 complex_edge_THFEM/emdefs.h File Reference

```
#include <cmath>
#include <complex>
```

Macros

- #define PI (4.0∗atan(1.0))
  vacuum magnetic permeability

Typedefs

- typedef double valtype
- typedef std::complex< valtype > cvaltype
- typedef nVector< vector_length, double > vec_type
- typedef nVector< vector_length, cvaltype > cvec_type
  vector_value type
- typedef valtype emtype
- typedef valtype(∗ Func) (const double ∗)
- typedef cvaltype(∗ CFunc) (const double ∗)
- typedef vec_type(∗ vFunc) (const double ∗)
- typedef cvec_type(∗ CvFunc) (const double ∗)

Functions

- const cvaltype I (0, 1)

Variables

- const double C =300
- const double eps0 =1./C
  speed of light ,nm/fs
- const double mue0 =1./C
  vacuum electrical permittivity

6.7.1 Macro Definition Documentation

6.7.1.1 PI

```
#define PI (4.0*atan(1.0))
```

vacuum magnetic permeability

6.7.2 Typedef Documentation

6.7.2.1 CFunc

```
typedef cvaltype(* CFunc) (const double *)
```

6.7.2.2 cvaltype

```
typedef std::complex<valtype> cvaltype
```

6.7.2.3 cvec_type

```
typedef nVector<vector_length,cvaltype> cvec_type
```

vector_value type

6.7.2.4 CvFunc

```
typedef cvec_type(* CvFunc) (const double *)
```

6.7.2.5 emtype

```
typedef valtype emtype
```

6.7.2.6 Func

```
typedef valtype(* Func) (const double *)
```

6.7.2.7 valtype

```
typedef double valtype
```

6.7.2.8 vec_type

```
typedef nVector<vector_length,double> vec_type
```

6.7.2.9 vFunc

```
typedef vec_type(* vFunc) (const double *)
```

6.7.3 Function Documentation

6.7.3.1 I()

```
const cvaltype I (
            0 ,
            1  )
```

6.7.4 Variable Documentation

6.7.4.1 C

```
const double C =300
```

6.7.4.2 eps0

```
const double eps0 =1./C
```

speed of light ,nm/fs

6.7.4.3 mue0

```
const double mue0 =1./C
```

vacuum electrical permittivity

6.8 complex_node_THFEM/emdefs.h File Reference

```
#include <cmath>
#include <complex>
```

Macros

- #define PI (4.0∗atan(1.0))
  vacuum magnetic permeability

Typedefs

- typedef double valtype
- typedef std::complex< valtype > cvaltype
- typedef valtype emtype
- typedef double(∗ Func) (const double ∗)
- typedef cvaltype(∗ CFunc) (const double ∗)

Functions

- const cvaltype I (0, 1)

Variables

- const double C =300
- const double eps0 =1./C
  speed of light ,nm/fs
- const double mue0 =1./C
  vacuum electrical permittivity

6.8.1 Macro Definition Documentation

6.8.1.1 PI

```
#define PI (4.0*atan(1.0))
```

vacuum magnetic permeability

6.8.2 Typedef Documentation

6.8.2.1 CFunc

```
typedef cvaltype(* CFunc) (const double *)
```

6.8.2.2 cvaltype

```
typedef std::complex<valtype> cvaltype
```

6.8.2.3 emtype

```
typedef valtype emtype
```

6.8.2.4 Func

```
typedef double(* Func) (const double *)
```

6.8.2.5 valtype

```
typedef double valtype
```

6.8.3 Function Documentation

6.8.3.1 I()

```
const cvaltype I (
             0 ,
             1  )
```

6.8.4 Variable Documentation

6.8.4.1 C

```
const double C =300
```

6.8.4.2 eps0

```
const double eps0 =1./C
```

speed of light ,nm/fs

6.8.4.3 mue0

```
const double mue0 =1./C
```

vacuum electrical permittivity

6.9 complex_edge_THFEM/main.cpp File Reference

```
#include "uiexp.h"
```

Macros

- #define EMUNITS

Functions

- int main (int argc, char ∗argv[ ])

6.9.1 Macro Definition Documentation

6.9.1.1 EMUNITS

```
#define EMUNITS
```

6.9.2 Function Documentation

6.9.2.1 main()

```
int main (
            int argc,
            char * argv[] )
```

6.10 complex_node_THFEM/main.cpp File Reference

```
#include "uiexp.h"
```

Macros

- #define EMUNITS

Functions

- int main (int argc, char ∗argv[ ])

6.10.1 Macro Definition Documentation

6.10.1.1 EMUNITS

```
#define EMUNITS
```

6.10.2 Function Documentation

6.10.2.1 main()

```
int main (
            int argc,
            char * argv[] )
```

6.11 complex_edge_THFEM/parameter.h File Reference

```
#include "emdefs.h"
```

Functions

- const vec_type val_0 (0)
- cvec_type u_exact (const double *p)
- cvaltype a (const double *p)
- cvaltype c (const double *p)
- cvec_type f (const double *p)
- cvaltype q (const double *p)
- cvec_type g1 (const double *p)
- cvec_type g2 (const double *p)
- cvec_type g3 (const double *p)
- cvec_type g4 (const double *p)
- cvaltype bnd (const double *p)

Variables

- double lambda = 10.
- double freq = C/lambda
- double k0 = 2*PI*lambda

6.11.1 Function Documentation

6.11.1.1 a()

```
cvaltype a (
            const double * p )
```

6.11.1.2 bnd()

```
cvaltype bnd (
            const double * p )
```

6.11.1.3 c()

```
cvaltype c (
            const double * p )
```

6.11.1.4 f()

```
cvec_type f (
            const double * p )
```

6.11.1.5 g1()

```
cvec_type g1 (
            const double * p )
```

6.11.1.6 g2()

```
cvec_type g2 (
            const double * p )
```

6.11.1.7 g3()

```
cvec_type g3 (
            const double * p )
```

6.11.1.8 g4()

```
cvec_type g4 (
            const double * p )
```

6.11.1.9 q()

```
cvaltype q (
            const double * p )
```

6.11.1.10 u_exact()

```
cvec_type u_exact (
            const double * p )
```

6.11.1.11 val_0()

```
const vec_type val_0 (
            0  )
```

6.11.2 Variable Documentation

6.11.2.1 freq

```
double freq = C/lambda
```

6.11.2.2 k0

```
double k0 = 2*PI*lambda
```

6.11.2.3 lambda

```
double lambda = 10.
```

6.12 complex_node_THFEM/parameter.h File Reference

```
#include "emdefs.h"
```

Functions

- cvaltype u_exact (const double *p)
- cvaltype a (const double *p)
- cvaltype c (const double *p)
- cvaltype f (const double *p)
- cvaltype q (const double *p)
- cvaltype g1 (const double *p)
- cvaltype g2 (const double *p)
- cvaltype g3 (const double *p)
- cvaltype g4 (const double *p)
- cvaltype bnd (const double *p)

Variables

- double lambda = 10.
- double freq = C/lambda
- double k0 = 2*PI*lambda

6.12.1 Function Documentation

6.12.1.1 a()

```
cvaltype a (
            const double * p )
```

6.12.1.2 bnd()

```
cvaltype bnd (
            const double * p )
```

6.12.1.3 c()

```
cvaltype c (
            const double * p )
```

6.12.1.4 f()

```
cvaltype f (
            const double * p )
```

6.12.1.5 g1()

```
cvaltype g1 (
             const double * p )
```

6.12.1.6 g2()

```
cvaltype g2 (
             const double * p )
```

6.12.1.7 g3()

```
cvaltype g3 (
             const double * p )
```

6.12.1.8 g4()

```
cvaltype g4 (
             const double * p )
```

6.12.1.9 q()

```
cvaltype q (
             const double * p )
```

6.12.1.10 u_exact()

```
cvaltype u_exact (
             const double * p )
```

6.12.2 Variable Documentation

6.12.2.1 freq

```
double freq = C/lambda
```

6.12.2.2 k0

```
double k0 = 2*PI*lambda
```

6.12.2.3 lambda

```
double lambda = 10.
```

6.13 complex_edge_THFEM/uiexp.cpp File Reference

```
#include "uiexp.h"
#include "parameter.h"
```

6.14 complex_node_THFEM/uiexp.cpp File Reference

```
#include "uiexp.h"
#include "parameter.h"
```

Functions

- void writeMatlabData (const std::string &filename, FEMFunction< double, DIM > u_h)

6.14.1 Function Documentation

6.14.1.1 writeMatlabData()

```
void writeMatlabData (
            const std::string & filename,
            FEMFunction< double, DIM > u_h )
```

6.15 complex_edge_THFEM/uiexp.h File Reference

```
#include <AFEPack/HGeometry.h>
#include <AFEPack/Geometry.h>
#include <AFEPack/TemplateElement.h>
#include <AFEPack/FEMSpace.h>
#include <AFEPack/DGFEMSpace.h>
#include <AFEPack/Operator.h>
#include <AFEPack/Functional.h>
#include <AFEPack/EasyMesh.h>
#include <AFEPack/MovingMesh2D.h>
#include <Eigen/Sparse>
#include <string>
#include <iostream>
#include <cstdlib>
#include <sstream>
#include <cmath>
#include "emdefs.h"
#include "datacache.h"
```

Classes

- class uiExperiment

Typedefs

- typedef Eigen::Triplet< cvaltype > T

Variables

- const int DIM = 2
- const int DOW = DIM
- const int TDIM = DIM
- const int vector_length = DIM

6.15.1 Typedef Documentation

6.15.1.1 T

```
typedef Eigen::Triplet<cvaltype> T
```

6.15.2 Variable Documentation

6.15.2.1 DIM

```
const int DIM = 2
```

6.15.2.2 DOW

```
const int DOW = DIM
```

6.15.2.3 TDIM

```
const int TDIM = DIM
```

6.15.2.4 vector_length

```
const int vector_length = DIM
```

6.16 complex_node_THFEM/uiexp.h File Reference

```
#include <AFEPack/HGeometry.h>
#include <AFEPack/Geometry.h>
#include <AFEPack/TemplateElement.h>
#include <AFEPack/FEMSpace.h>
#include <AFEPack/DGFEMSpace.h>
#include <AFEPack/Operator.h>
#include <AFEPack/Functional.h>
#include <AFEPack/EasyMesh.h>
#include <AFEPack/MovingMesh2D.h>
#include <Eigen/Sparse>
#include <string>
#include <iostream>
#include <cstdlib>
#include <sstream>
#include <cmath>
#include "emdefs.h"
#include "datacache.h"
```

Classes

- class uiExperiment

Typedefs

- typedef Eigen::Triplet< cvaltype > T

Variables

- const int DIM = 2
- const int DOW = DIM
- const int TDIM = DIM

6.16.1 Typedef Documentation

6.16.1.1 T

```
typedef Eigen::Triplet<cvaltype> T
```

6.16.2 Variable Documentation

6.16.2.1 DIM

```
const int DIM = 2
```

6.16.2.2 DOW

```
const int DOW = DIM
```

6.16.2.3 TDIM

```
const int TDIM = DIM
```

6.17 mainpage.h File Reference

Index