User's Guide*

François Cuvelier†

December 16, 2017

Abstract

This Octave package uses a SIMESH object, comming from the FC-SIMESH package, to display simplicial meshes or datas on simplicial meshes. Its kernel uses the FC-GRAPHICS4MESH package.

0 Contents

1 Introduction **2**

2 Installation **2**

2.1 Installation automatic, all in one (recommanded) 2

3 Mesh **5**

4 PLOTMESH function **5**

5 PLOT function **10**

6 PLOTISO function **14**

7 SLICEMESH function **18**

8 SLICE function **19**

9 SLICEISO function **20**

*Compiled with Octave 4.2.1

†Université Paris 13, Sorbonne Paris Cité, LAGA, CNRS UMR 7539, 99 Avenue J-B Clément, F-93430 Villetaneuse, France, cuvelier@math.univ-paris13.fr.

This work was supported by the ANR project DEDALES under grant ANR-14-CE23-0005.

1 Introduction

This **experimental** Octave package uses the Simplicial meshes could be:

- a triangular mesh in dimension 2, made with 2-simplices (ie. triangles),
- a tetrahedral mesh in dimension 3, made with 3-simplices (ie. tetrahedron),
- a triangular mesh in dimension 3 (surface mesh), made with 2-simplices,
- a line mesh in dimension 2 or 3 made with 1-simplices (ie. lines).

A simplicial mesh is given by its vertices array q and its connectivity array me . For demonstration purpose, some simplicial meshes are given in this package. They can be load by using the function getMesh2D, getMesh3D or getMesh3Ds of the fc_graphics4mesh package.

This package was tested under

Windows 10.0.16299: with Octave 4.2.0 and 4.2.1 ('fltk' graphics toolkit, NVIDIA Quadro K600 with nvidia-376 driver)

Mac OS X 10.12.6: with Octave 4.2.1 (installed with homebrew, 'fltk' graphics toolkit, Intel HD Graphics 3000 with 2.1 INTEL-10.2.37 driver)

Ubuntu 14.04.5 LTS: with Octave 4.2.0 and 4.2.1 (both compiled from source, 'fltk' and 'qt' graphics toolkits, NVIDIA Quadro K1100M with nvidia-340 driver)

Ubuntu 16.04.3 LTS: with Octave 4.2.0 and 4.2.1 (both compiled from source, 'fltk' and 'qt' graphics toolkits, NVIDIA GeForce GTX 1070 with nvidia-384 driver)

Ubuntu 17.10: with Octave 4.2.0 and 4.2.1 (both compiled from source, NVIDIA GeForce GTX 1080 Ti with nvidia-384 driver)

It is not compatible with Octave 4.0.x and previous.

2 Installation

2.1 Installation automatic, all in one (recommanded)

For this method, one just have to get/download the install file

ofc_siplt_install.m

or get it on the dedicated web page. Thereafter, one run it under Octave. This command download, extract and configure the *fc-siplt* and the required *fc-tools* package in the current directory.

For example, to install this package in `~/Octave/packages` directory, one have to copy the file mfc_siplt_install.m in the `~/Octave/packages` directory. Then in a Octave terminal run the following commands

```
>> cd ~/Octave/packages
>> mfc_siplt_install
```

There is the output of the `mfc_siplt_install` command on a Linux computer:

```
Parts of the Octave <fc-siplt> package.
Copyright (C) 2017 F. Cuvelier <cuvelier@math.univ-paris13.fr>

************************************
Downloading and installing the package
    <fc-simesh>[0.2.1]
************************************
Parts of the GNU Octave <fc-simesh> package.
Copyright (C) 2016-2017 Francois Cuvelier <cuvelier@math.univ-paris13.fr>

1- Downloading and extracting the packages
   -> <fc-tools>[0.0.19] ... OK
   -> <fc-hypermesh>[0.0.6] ... OK
   -> <fc-oogmsh>[0.0.17] ... OK
   -> <fc-simesh>[0.2.1] ... OK
   -> <fc-graphics4mesh>[0.0.2] ... OK
   -> <fc-siplt>[0.0.2] ... OK
2- Setting the packages
2-a) Setting the <fc-hypermesh> package
Write in ...
     /home/cuvelier/Octave/packages/fc-siplt-full/fc_hypermesh-0.0.6/configure_loc.m ...
     ...
  -> done
2-b) Setting the <fc-oogmsh> package
[fc-oogmsh] Using GMSH binary : /home/cuvelier/bin/gmsh
[fc-oogmsh] Writing in ...
     /home/cuvelier/Octave/packages/fc-siplt-full/fc_oogmsh-0.0.17/configure_loc.m ...
     ...
[fc-oogmsh] configured with
   ->      gmsh_bin='/home/cuvelier/bin/gmsh';
   ->      ...
        mesh_dir='/home/cuvelier/Octave/packages/fc-siplt-full/fc_oogmsh-0.0.17/meshes';
   ->       ...
        geo_dir='/home/cuvelier/Octave/packages/fc-siplt-full/fc_oogmsh-0.0.17/geodir';
   -> ...
        fc_tools_dir='/home/cuvelier/Octave/packages/./fc-siplt-full/fc_tools-0.0.19';
[fc-oogmsh] done
2-c) Setting the <fc-simesh> package without graphics
[fc-simesh] Unable to load the fc-siplt toolbox/package in current path
[fc-simesh] Guess path does not exists:
   -> siplt
[fc-] Guess path does not exists:
   -> [fc-simesh] Use fc_simesh.configure('fc_siplt_dir',<DIR>) to ...
        correct this issue

[fc-simesh] no graphics package installed
[fc-simesh] Writing in ...
     /home/cuvelier/Octave/packages/fc-siplt-full/fc_simesh-0.2.1/configure_loc.m ...
     ...
[fc-simesh] configured with
   -> oogmsh_dir    ...
        ='/home/cuvelier/Octave/packages/./fc-siplt-full/fc_oogmsh-0.0.17';
   -> hypermesh_dir ...
        ='/home/cuvelier/Octave/packages/./fc-siplt-full/fc_hypermesh-0.0.6';
   -> siplt_dir     ='';
[fc-simesh] done
2-d) Setting the <fc-graphics4mesh> toolbox
Write in ...
     /home/cuvelier/Octave/packages/fc-siplt-full/fc_graphics4mesh-0.0.2/configure_loc.m ...
     ...
  -> done
2-e) Setting the <fc-siplt> toolbox
Write in ...
     /home/cuvelier/Octave/packages/fc-siplt-full/fc_siplt-0.0.2/configure_loc.m ...
     ...
  -> done
2-f) Setting the <fc-simesh> toolbox with graphics
[fc-simesh] Writing in ...
     /home/cuvelier/Octave/packages/fc-siplt-full/fc_simesh-0.2.1/configure_loc.m ...
     ...
[fc-simesh] configured with
   -> oogmsh_dir    ...
        ='/home/cuvelier/Octave/packages/./fc-siplt-full/fc_oogmsh-0.0.17';
   -> hypermesh_dir ...
        ='/home/cuvelier/Octave/packages/./fc-siplt-full/fc_hypermesh-0.0.6';
   -> siplt_dir     ...
        ='/home/cuvelier/Octave/packages/./fc-siplt-full/fc_siplt-0.0.2';
[fc-simesh] done
3- Using instructions
   To use the <fc-simesh> package:
   addpath('/home/cuvelier/Octave/packages/./fc-siplt-full/fc_simesh-0.2.1')
   fc_simesh.init()

   See /home/cuvelier/Octave/packages/ofc_simesh_set.m
    <fc-simesh>[0.2.1]: installed

*** Using instructions
   To use the <fc-siplt> package:
   addpath('/home/cuvelier/Octave/packages/./fc-siplt-full/fc_simesh-0.2.1')
```

The complete package (i.e. with all the other needed packages) is stored in the directory `~/Octave/packages/fc-siplt-full` and, for each Octave session, one have to set the package by:

```
>> addpath('~/Octave/packages/fc-siplt-full/fc_simesh-0.2.1')
>> fc_siplt.init()
```

For **uninstalling**, one just have to delete directory

`~/Octave/packages/fc-siplt-full`

3 Mesh

The functions getMesh2D, getMesh3D and getMesh3Ds return a mesh vertices array q, a mesh elements connectivity array associated with the input argument d (simplex dimension) and the indices array toGlobal. The vertices array q is a dim-by-n_q array where dim is the space dimension (2 or 3) and n_q the number of vertices. The connectivity array me is a $(d+1)$-by-n_{me} array where n_{me} is the number of mesh elements and $0 \leqslant d \leqslant dim$ is the simplicial dimension:

- $d = 0$: points,
- $d = 1$: lines,
- $d = 2$: triangle,
- $d = 3$: tetrahedron.

So we can use theses functions to obtain

- 3D mesh: getMesh3D(3) (*main* mesh), getMesh3D(2), getMesh3D(1), getMesh3D(0),
- 3D surface mesh: getMesh3Ds(2) (*main* mesh), getMesh3Ds(1) , getMesh3Ds(0),
- 2D mesh: getMesh2D(2) (*main* mesh), getMesh2D(1), getMesh2D(0).

For example,

- [q3,me3,toGlobal3]=fc_graphics4mesh.getMesh3D(3) return a 3-simplicial mesh (main mesh) in space dimension $dim = 3$,
- [q2,me2,toGlobal2]=fc_graphics4mesh.getMesh3D(2) return a 2-simplicial mesh in space dimension $dim = 3$.

The third output are indices of the vertices in the *main* mesh:
q3(:,toGlobal2) == q2

4 PLOTMESH function

The method PLOTMESH displays the mesh or parts of the mesh defined by an SIMESH object.

Syntaxe

```
fc_siplt.plotmesh(Th,)
fc_siplt.plotmesh(Th,Name,Value, ...)
```

Description

`fc_siplt.plotmesh(Th,)` displays all the Th.d-dimensional simplices elements.

`fc_siplt.plotmesh(Th,Name,Value, ...)` specifies function options using one or more Name,Value pair arguments. Options of first level are

- `'d'` : to specify the dimension of the simplices elements (default : Th.d)
- `'labels'` : to select the labels of the elements to display,
- `'color'` : to specify the color of the displayed mesh elements. (default : use one color by displayed mesh elements),
- `'inlegend'` : add a legend name to graph if true (default : false)
- `'bounds'` : If true, draw the borders of the selected elementaries mesh elements (only for 2-dimensional simplices). (default : false)
- `'cutPlan'` : cut mesh by n plans given by n-by-4 array P where the equation of the i-th cut plan is given by

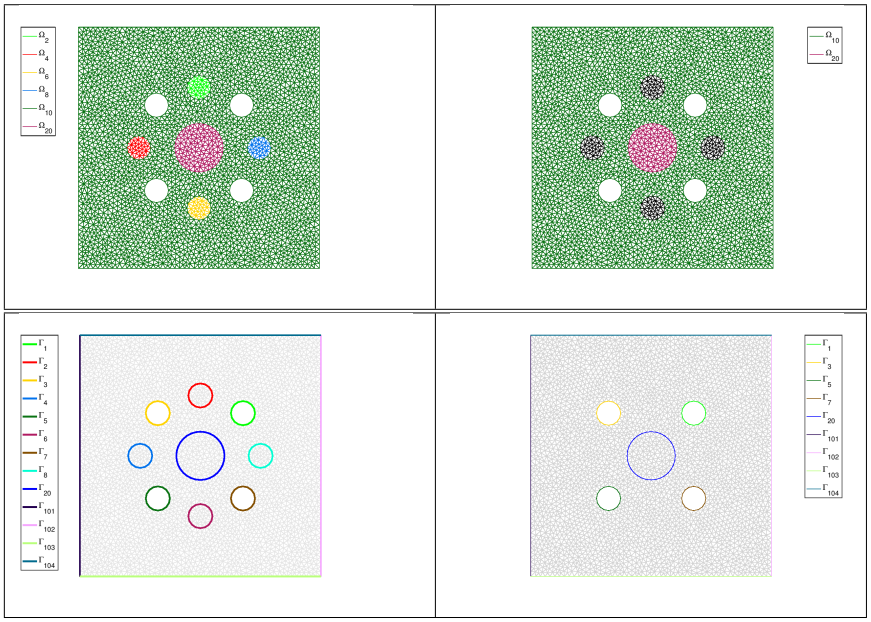
$$P(i,1)\,x + P(i,2)\,y + P(i,3)\,z + P(i,4) = 0.$$

  The normal vector $P(i,1:3)$ pointed to the part of the mesh not displayed. (only for simplices in dimension 3) default : [] (no cut).

The options of second level depend on the type of elementaries mesh elements to represent.
One can use any option of the following functions according to the type of d-simplex to be represented.

- In dimension 3,
  - if $d == 3$, **patch** function is used,
  - if $d == 2$, trimesh function is used,
  - if $d == 1$, **plot3** function is used,
  - if $d == 0$, **plot3** function is used,
- In dimension 2,
  - if $d == 2$, trimesh function is used,
  - if $d == 1$, **plot** function is used,
  - if $d == 0$, **plot** function is used,
- In dimension 1,
  - if $d == 1$, **line** function is used,
  - if $d == 0$, **plot** function is used,

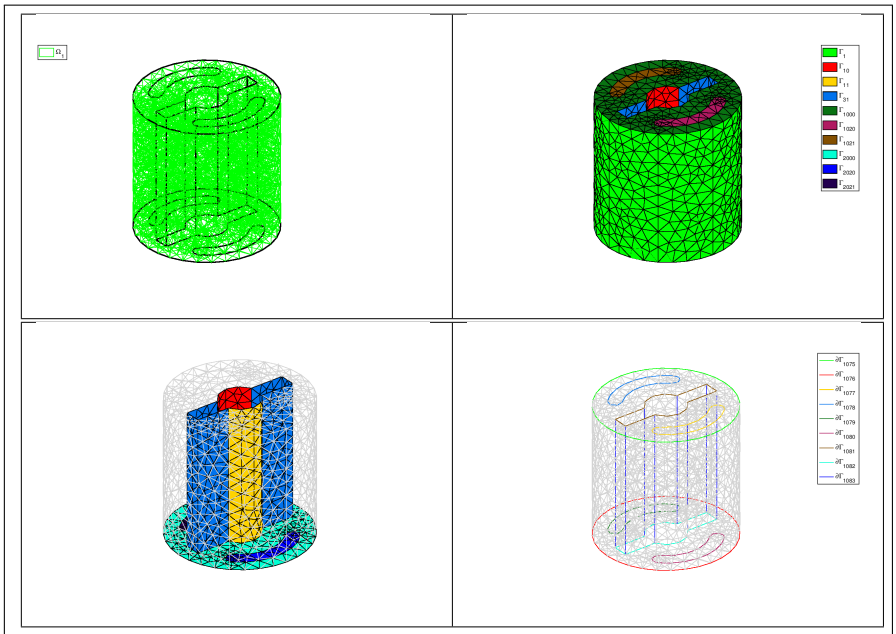
2D example The following example use the *.geo* file `condenser11.geo` which is in the directory `geodir` of the toolbox ....



```
meshfile=gmsh.buildmesh2d('condenser11',25,'verbose',0);
Th=siMesh(meshfile);
figure(1)
fc_siplt.plotmesh(Th,'inlegend',true)
axis off;axis image;
set(legend('show'),'Location','NorthWestOutside','fontsize',12)
figure(2)
fc_siplt.plotmesh(Th,'labels',[10,20],'inlegend',true)
hold on;axis off;axis image
fc_siplt.plotmesh(Th,'labels',[2,4,6,8],'Color','k','Linestyle',':')
set(legend('show'),'Location','NorthEastOutside','fontsize',12)
figure(3)
fc_siplt.plotmesh(Th,'Color',0.9*[1,1,1])
hold on;axis off;axis image;
fc_siplt.plotmesh(Th,'d',1,'inlegend',true,'LineWidth',2)
set(legend('show'),'Location','NorthWestOutside','fontsize',12)
figure(4)
fc_siplt.plotmesh(Th,'Color','LightGray')
hold on;axis off;axis image
fc_siplt.plotmesh(Th,'d',1,'inlegend',true,'labels',[1:2:7,20,101:104])
set(legend('show'),'Location','NorthEastOutside','fontsize',12)
```

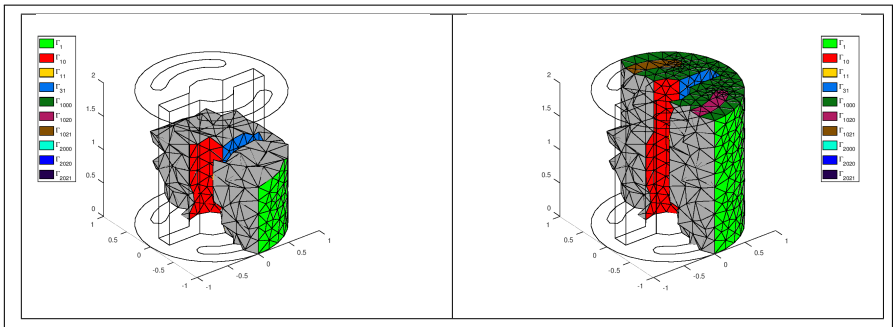
Listing 1: 2D plot mesh

3D example The following example use the *.geo* file `cylinderkey.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5,'verbose',0);
Th=siMesh(meshfile);
figure(1)
fc_siplt.plotmesh(Th,'inlegend',true)
hold on;axis off;axis image
fc_siplt.plotmesh(Th,'d',1,'Color','k','Linewidth',1.5)
set(legend('show'),'Location','NorthWestOutside')
figure(2)
fc_siplt.plotmesh(Th,'d',2,'inlegend',true)
view(3);hold on;axis off;axis image
set(legend('show'),'Location','NorthEastOutside')
figure(3)
fc_siplt.plotmesh(Th,'d',2,'labels',[1,1000,1020,1021], 'EdgeColor','LightGray', ...
    'EdgeAlpha',0.4,'FaceColor','none')
hold on;axis off;axis image
%fc_siplt.plotmesh(Th,'d',2,'labels',1000,'bounds',true,'color','k')
fc_siplt.plotmesh(Th,'d',2, 'labels',[10,11,31,2000,2020,2021])
figure(4)
fc_siplt.plotmesh(Th,'d',2, 'EdgeColor','LightGray', ...
    'EdgeAlpha',0.4,'FaceColor','none')
hold on;axis off;axis image
fc_siplt.plotmesh(Th,'d',1,'inlegend',true)
set(legend('show'),'Location','NorthEastOutside')
```

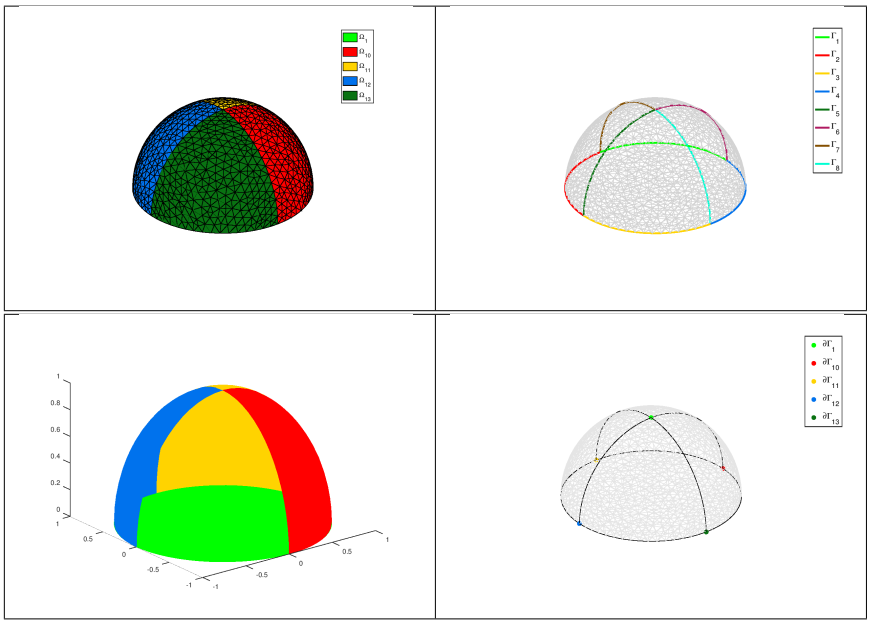
Listing 2: 3D plot mesh



```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5,'verbose',0);
figure(5)
P=[fc_tools.graphics.PlaneCoefs([0 0 1], [0 0 1]); ...
   fc_tools.graphics.PlaneCoefs([0 0 1],[-1 0 0])];
fc_siplt.plotmesh(Th,'d',1,'color','black')
hold on;axis image;view(-75,36)
fc_siplt.plotmesh(Th,'cutPlan',P,'Color','DarkGrey')
fc_siplt.plotmesh(Th,'d',2,'cutPlan',P,'inlegend',true);
set(legend('show'),'Location','NorthWestOutside')
figure(6)
P=fc_tools.graphics.PlaneCoefs([0 0 1],[-1 0 0]);
fc_siplt.plotmesh(Th,'d',1,'color','black')
hold on;axis image;view(-75,36)
fc_siplt.plotmesh(Th,'cutPlan',P,'Color','DarkGrey')
fc_siplt.plotmesh(Th,'d',2,'cutPlan',P,'inlegend',true);
set(legend('show'),'Location','NorthEastOutside')
```

Listing 3: 3D plot mesh,

3D surface example The following example use the *.geo* file `demisphere5.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D surface mesh with simplices of dimensions 1 and 2.



```
meshfile=fc_oogmsh.buildmesh3ds('demisphere5',ns*5,'verbose',0);
Th=siMesh(meshfile);
figure(1)
fc_siplt.plotmesh(Th,'inlegend',true)
axis off;axis equal;legend('show')
figure(2)
fc_siplt.plotmesh(Th,'EdgeColor','LightGray', 'EdgeAlpha',0.4,'FaceColor','none')
view(3);hold on;axis off;axis equal
fc_siplt.plotmesh(Th,'d',1,'inlegend',true,'LineWidth',2)
set(legend('show'),'Location','NorthEastOutside','FontSize',12)
figure(3)
fc_siplt.plotmesh(Th,'labels',[1,10,11,12],'EdgeColor','none')
%hold on;axis off;axis equal
%fc_siplt.plotmesh(Th,'labels',13,'bounds',true,'color','k','LineWidth',2)
figure(4)
fc_siplt.plotmesh(Th,'EdgeColor',0.9*[1,1,1], 'EdgeAlpha',0.4,'FaceColor','none')
hold on;axis off;axis equal
fc_siplt.plotmesh(Th,'d',1,'color','k')
fc_siplt.plotmesh(Th,'d',0,'inlegend',true)
set(legend('show'),'Location','NorthEastOutside','FontSize',12)
```

Listing 4: 3D surface mesh : plot function

5 PLOT function

The method FC_SIPLT.PLOT displays scalar datas on the mesh or parts of the mesh defined by an SIMESH object.

Syntaxe

```
fc_siplt.plot(Th,u)
fc_siplt.plot(Th,u,Name,Value, ...)
```

Description

fc_siplt.**plot**(Th,u) displays data u on all the Th.d-dimensional simplices

elements. The data u is an 1D-array of size Th.nq or Th.nqGlobal or Th.nqParent.

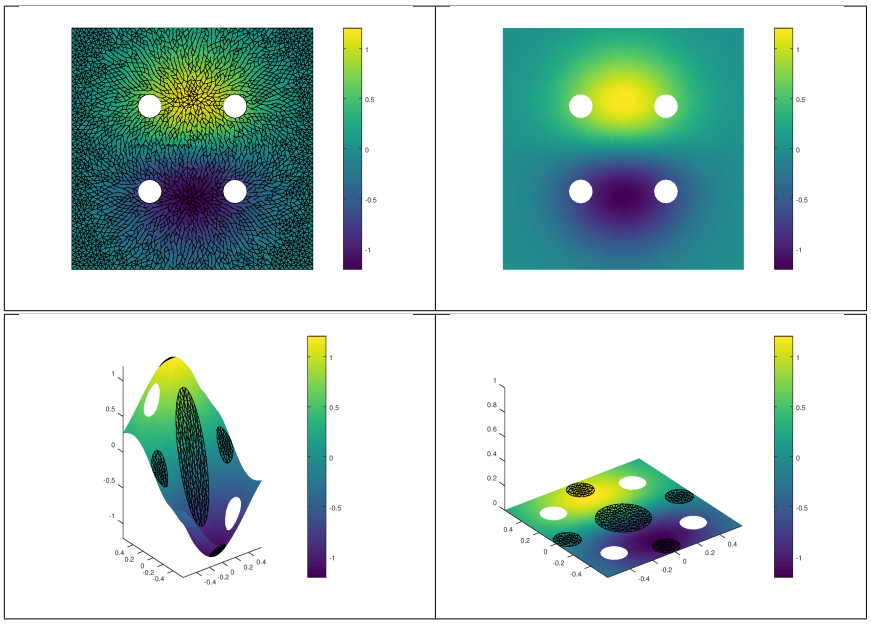
fc_siplt .**plot**(Th,u,Name,Value, ...) specifies function options using one or more Name,Value pair arguments. Options of first level are

- `'d'` : to specify the dimension of the simplices elements (default : Th.d)
- `'labels'` : to select the labels of the elements to display data,
- `'plan'` : if true, (default : false)

The options of second level depend on the type of elementaries mesh elements on which we want to represent datas.
One can use any option of the following functions according to the type of d-simplex.

- In dimension 3, **patch** function is used for $d \in [\![1, 3]\!]$.
- In dimension 2,
  - for $d == 2$, if `'plan'` is true, **patch** function is used, otherwise trisurf function,
  - for $d == 1$, **patch** function is used.
- In dimension 1 and $d == 1$, **plot** function is used

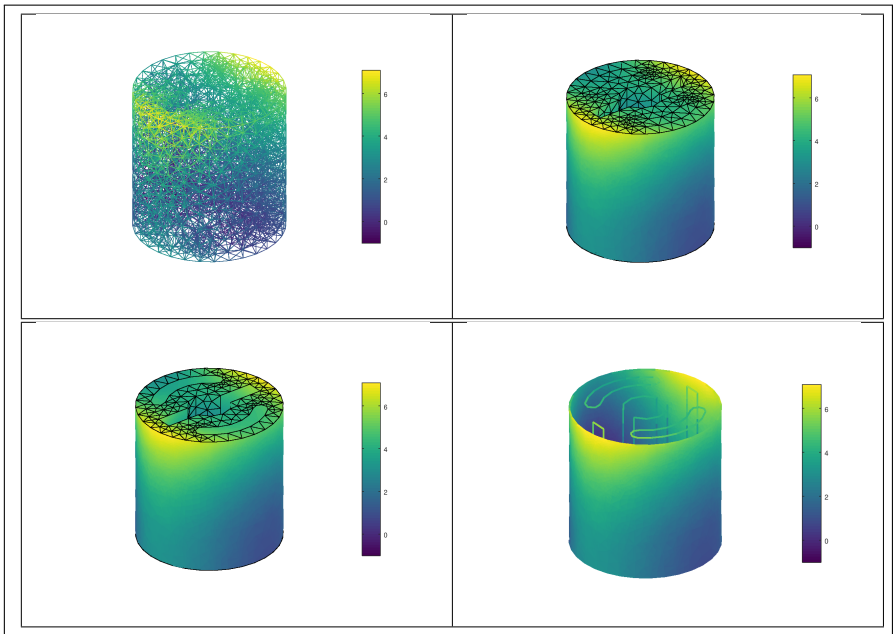
2D example The following example use the *.geo* file `condenser11.geo` which is in the directory `geodir` of the toolbox.



```
geofile=fc_simesh.get_geo(2,2,'condenser11');
meshfile=gmsh.buildmesh2d(geofile,25,'verbose',0);
Th=siMesh(meshfile);
u=Th.eval(@(x,y) 5*exp(-3*(x.^2+y.^2)).*cos(x).*sin(y));
figure(1)
fc_siplt.plot(Th,u)
axis off;axis image;colorbar
figure(2)
fc_siplt.plot(Th,u)
axis off;axis image;shading interp;colorbar
figure(3)
fc_siplt.plot(Th,u,'labels',[2:2:8,20],'FaceColor','interp')
view(3);hold on;axis image;colorbar
fc_siplt.plot(Th,u,'labels',10,'FaceColor','interp','EdgeColor','none')
figure(4)
fc_siplt.plot(Th,u,'labels',[2:2:8,20], 'plan',true ,'FaceColor','interp')
view(3);hold on;axis image;colorbar
fc_siplt.plot(Th,u,'labels',10, 'plan',true, ...
    'FaceColor','interp','EdgeColor','none')
```

Listing 5: 2D mesh : fc_siplt.**plot** function

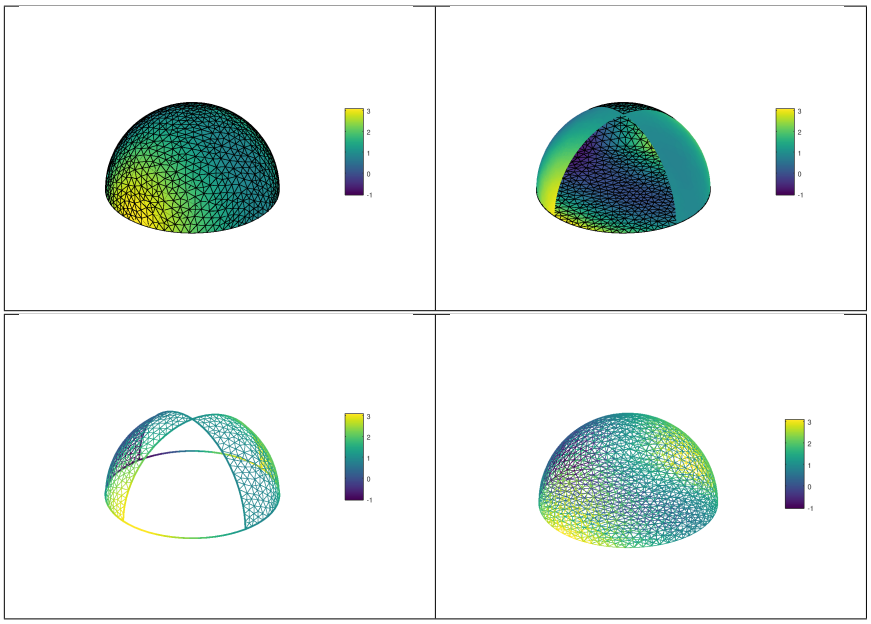
3D example The following example use the *.geo* file `cylinderkey.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5);
Th=siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plot(Th,u);
axis off;axis image;colorbar
figure(2)
fc_siplt.plot(Th,u,'d',2,'labels',[10,11,31,1000,1020,1021,2000,2020,2021])
hold on;axis off;axis image;colorbar
fc_siplt.plot(Th,u,'d',2,'labels',1,'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
figure(3)
fc_siplt.plot(Th,u,'d',2,'labels',1,'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
hold on;axis off;axis image;colorbar
fc_siplt.plot(Th,u,'d',2,'labels',[10,11,1000,2000])
fc_siplt.plot(Th,u,'d',2,'labels',[31,1020,1021,2020,2021], ...
    'FaceColor','interp','EdgeColor','none')
figure(4)
fc_siplt.plot(Th,u,'d',2,'labels',1,'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
hold on;axis off;axis image;colorbar
fc_siplt.plot(Th,u,'d',1,'LineWidth',2)
```

Listing 6: 3D mesh : fc_siplt.**plot** function

3D surface example The following example use the *.geo* file `demisphere5.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D surface mesh with simplices of dimensions 1 and 2.



```
geofile=fc_simesh.get_geo(3,2,'demisphere5');
meshfile=fc_oogmsh.buildmesh3ds(geofile,ns*5,'verbose',0);
Th=siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plot(Th,u)
axis off;axis image;colorbar;
figure(2)
fc_siplt.plot(Th,u,'labels',[1,11])
hold on;axis off;axis image;colorbar;
fc_siplt.plot(Th,u,'labels',[10,12], 'FaceColor','interp', 'EdgeColor','none')
figure(3)
fc_siplt.plot(Th,u,'d',1,'LineWidth',2)
hold on;axis off;axis image;colorbar
fc_siplt.plot(Th,u,'labels',[10,12], 'FaceColor','none', 'EdgeColor','interp')
figure(4)
fc_siplt.plot(Th,u,'FaceColor','none', 'EdgeColor','interp')
axis off;axis image;colorbar;
```

Listing 7: 3D surface mesh : fc_siplt.**plot** function

6 PLOTISO function

The function FC_SIPLT.PLOTISO displays isolines from datas on the mesh or parts of the mesh defined by an SIMESH object. This function only works with 2-simplices in space dimension 2 or 3.

Syntaxe

```
fc_siplt.plotiso(Th,u)
fc_siplt.plotiso(Th,u,Name,Value, ...)
```

Description

`fc_siplt.plotiso(Th,u)` displays data u on all the 2-dimensional simplices elements. The data u is an 1D-array of size Th.nq or Th.nqGlobal or Th.nqParent.

fc_siplt.plotiso(Th,u,key,value, ...) specifies function options using one or more key,value pair arguments. Options of first level are

- `'niso'` : to specify the number of isolines (default : 10)
- `'isorange'` : to specify the list of isovalues (default : empty)
- `'isocolorbar'` : if true, colorbar with isovalues is drawn (default : false)
- `'format'` : to specify the format of the isovalues on the colorbar (default : '%g')
- `'labels'` : to select the labels of the elements to display data,
- `'plan'` : if true, (default : false)
- `'color'` : to specify one color for all isolines (default : empty)
- `'mouse'` : if true, display information on clicked isoline (default : false)

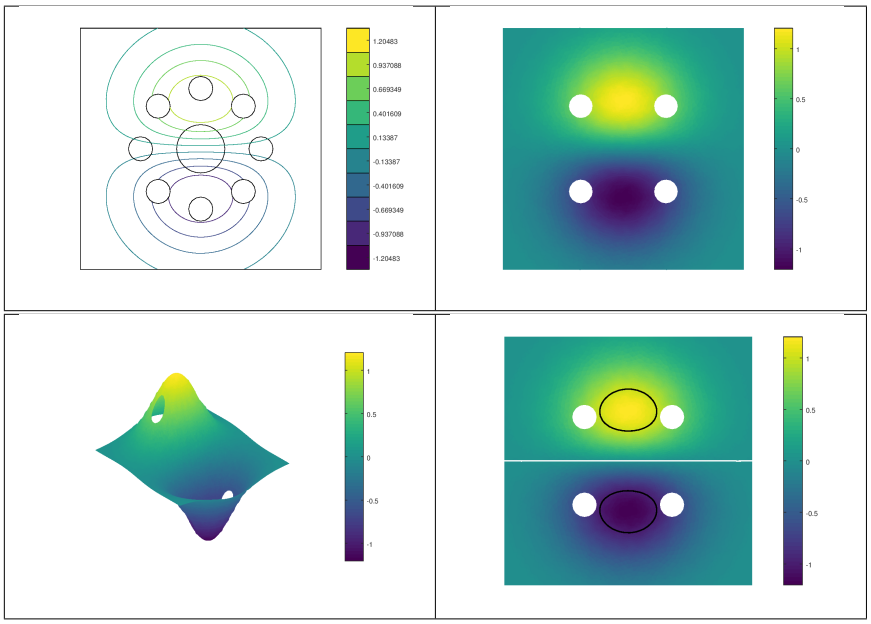
The options of second level are all options of

- **plot3** function in dimension 3 or in dimension 2 with `'plan'` set to false
- **plot** function in 2 with `'plan'` set to true

This function accepts until 4 output arguments :

- *bullet* 1st output is the colors of the isolines
- *bullet* 2nd output is the isovalues of the isolines
- *bullet* 3th output is the handle of the colobar iso.
- *bullet* 4th output is all the handles of the isolines as an 2D-array of dimension N-by-niso, where N is the number of 2-simplex elementary meshes where isolines are drawn.

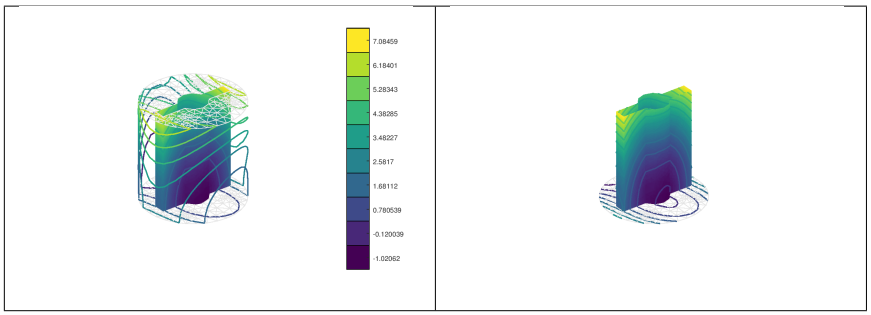
2D example The following example use the *.geo* file `condenser11.geo` which is in the directory `geodir` of the toolbox.



```
meshfile=gmsh.buildmesh2d('condenser11',25,'verbose',0);
Th=siMesh(meshfile);
u=Th.eval(@(x,y) 5*exp(-3*(x.^2+y.^2)).*cos(x).*sin(y));
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k')
hold on;axis off;axis image;
fc_siplt.plotiso(Th,u,'isocolorbar',true)
figure(2)
fc_siplt.plot(Th,u,'plan',true,'FaceAlpha',0.7)
hold on;axis off;axis image;shading interp;
fc_siplt.plotiso(Th,u,'plan',true,'LineWidth',1.5)
colorbar
figure(3)
fc_siplt.plot(Th,u,'FaceAlpha',0.7)
view(3)
shading interp;hold on;axis off;axis image;
fc_siplt.plotiso(Th,u,'niso',15,'LineWidth',1.5)
colorbar
figure(4)
fc_siplt.plot(Th,u,'plan',true)
shading interp;hold on;axis off;axis image;
fc_siplt.plotiso(Th,u,'isorange',0, 'LineWidth',1.5, 'color','w')
fc_siplt.plotiso(Th,u,'isorange',[-1,1], 'LineWidth',1.5, 'color','k','plan',true)
axis off;axis image;colorbar
```

Listing 8: 2D mesh : fc_siplt.plotiso function

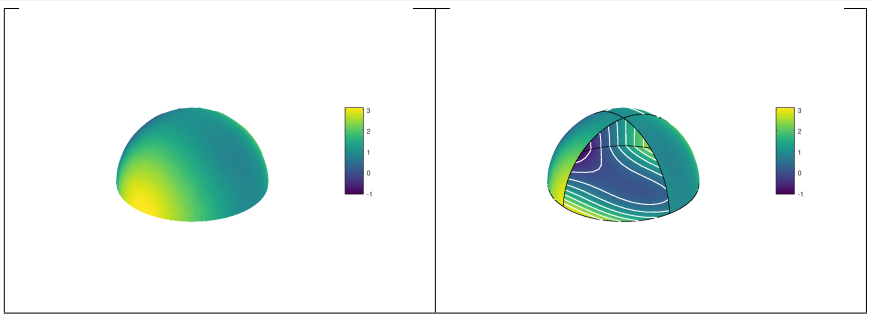
3D example The following example use the *.geo* file `cylinderkey.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.




```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=gmsh.buildmesh3d(geofile,ns*5,'verbose',0);
Th=siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plot(Th,u,'d',2,'labels',[10,11,31],'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
hold on;view(3);axis off;axis equal;
fc_siplt.plotmesh(Th,'d',2,'labels',[1000,1020,1021,2000,2020,2021], ...
    'FaceColor','none','EdgeColor',0.9*[1,1,1])
fc_siplt.plotiso(Th,u,'isocolorbar',true,'LineWidth',1.5)
figure(2)
fc_siplt.plot(Th,u,'d',2,'labels',[10,11,31],'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
hold on;axis off;axis equal;
fc_siplt.plotmesh(Th,'d',2,'labels',[2000,2020,2021], ...
    'FaceColor','none','EdgeColor',0.9*[1,1,1])
fc_siplt.plotiso(Th,u,'labels',[10,11,31,2000,2020,2021],'LineWidth',1.5, ...
    'niso',15)
```

Listing 9: 3D mesh : fc_siplt.plotiso function

3D surface example The following example use the *.geo* file `demisphere5.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D surface mesh with simplices of dimensions 1 and 2.




```
meshfile=gmsh.buildmesh3ds('demisphere5',ns*5,'verbose',0);
Th=siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plot(Th,u,'FaceColor','interp', 'EdgeColor','none','FaceAlpha',0.4)
hold on;axis off;axis equal;colorbar
fc_siplt.plotiso(Th,u,'LineWidth',1.5)
figure(2)
fc_siplt.plot(Th,u,'labels',[1,11],'FaceColor','interp', 'EdgeColor','none')
hold on;axis off;axis equal;colorbar;
fc_siplt.plotiso(Th,u,'labels',[1,11],'LineWidth',1.,'color','w')
fc_siplt.plot(Th,u,'labels',[10,12],'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
fc_siplt.plotiso(Th,u,'labels',[10,12],'LineWidth',1.5)
fc_siplt.plotmesh(Th,'d',1,'Color','k')
```

Listing 10: 3D surface mesh : fc_siplt.plotiso function

7 SLICEMESH function

The function FC_SIPLT.SLICEMESH displays intersection of a plane and a 3D mesh or parts of a 3D mesh defined by an SIMESH object.

Syntaxe

```
fc_siplt.slicemesh(Th,P)
fc_siplt.slicemesh(Th,P,Name,Value, ...)
```

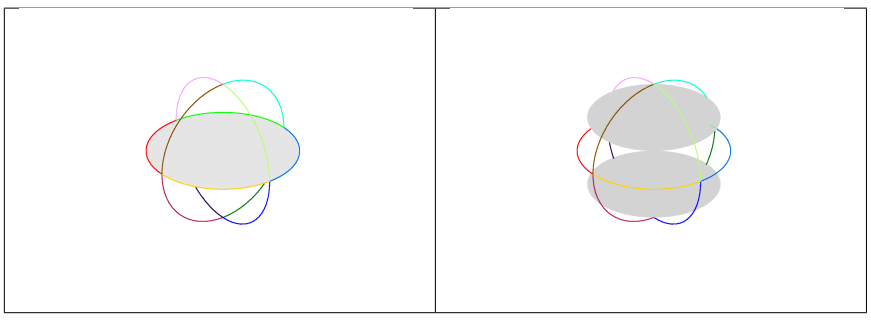
Description

`fc_siplt.slicemesh(Th,P)` displays intersection of the plane defined by $P(1)x+P(2)y+P(3)z+P(4)=0$ and all the 3-dimensional simplices elements. To compute P one can use the function PlaneCoefs of the FC-SIPLT package. With this function, the array P, is obtained with P=PlaneCoefs(Q,V) where Q is a point in the plane and V is a vector orthogonal to it.

`fc_siplt.plot(Th,u,Name,Value, ...)` specifies function options using one or more Name,Value pair arguments. Options of first level are

- `'color'` : to specify the slice color (default : light grey, rgb=[0.9,0.9,0.9])
- `'labels'` : to select the labels of the elements to intersect,

3D example The following example use the *.geo* file `ball8.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
meshfile=fc_oogmsh.buildmesh3d('ball8',ns*10,'verbose',0);
Th=siMesh(meshfile);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'LineWidth',1)
hold on;axis off;axis image;
P=fc_tools.graphics.PlaneCoefs([0 0 0],[0 0 1]);
fc_siplt.slicemesh(Th,P,'color',0.9*[1 1 1])
figure(2)
P=[fc_tools.graphics.PlaneCoefs([0 0 -0.5],[0 0 1]); ...
   fc_tools.graphics.PlaneCoefs([0 0 0.5],[0 0 1])];
fc_siplt.plotmesh(Th,'d',1,'LineWidth',1);
hold on;axis off;axis image;
fc_siplt.slicemesh(Th,P,'Color','LightGray')
```

Listing 11: 3D mesh : slicemesh function

8 SLICE function

The method FC_SIPLT.SLICE displays datas on the intersection of a plane and a 3D mesh or parts of a 3D mesh defined by an SIMESH object.

Syntaxe

```
fc_siplt.slice(Th,u,P)
fc_siplt.slice(Th,u,P,Name,Value, ...)
```

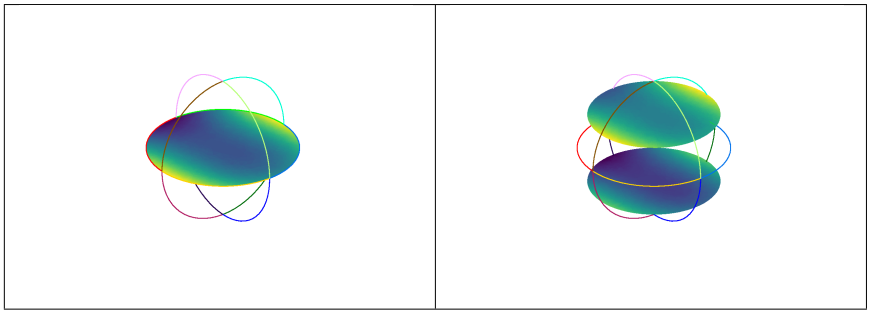
Description

fc_siplt.**slice**(Th,u,P) displays u data on the intersection of the plane defined by $P(1)x + P(2)y + P(3)z + P(4) = 0$ and all the 3-dimensional simplices elements. The data u is an 1D-array of size Th.nq or Th.nqGlobal or Th.nqParent. To compute P one can use the function PLANECOEFS of the FC-TOOLS package. With this function, the array P, is obtained with P=PlaneCoefs(Q,V) where Q is a point in the plane and V is a vector orthogonal to it.

Th.slide(u,P,Name,Value, ...) specifies function options using one or more Name,Value pair arguments. Options of first level are

- `'labels'` : to select the labels of the elements to intersect,

3D example The following example use the *.geo* file `ball8.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
meshfile=fc_oogmsh.buildmesh3d('ball8',ns*10,'verbose',0);
Th=siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y+z);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'LineWidth',1)
hold on;axis off;axis image;
P=fc_tools.graphics.PlaneCoefs([0 0 0],[0 0 1]);
fc_siplt.slice(Th,u,P,'Facecolor','interp')
figure(2)
P=[fc_tools.graphics.PlaneCoefs([0 0 -0.5],[0 0 1]); ...
   fc_tools.graphics.PlaneCoefs([0 0 0.5],[0 0 1])];
fc_siplt.plotmesh(Th,'d',1,'LineWidth',1);
hold on;axis off;axis image;
fc_siplt.slice(Th,u,P,'Facecolor','interp')
```

Listing 12: 3D mesh : **slice** function

9 SLICEISO function

The method FC_SIPLT.SLICEISO displays isolines of data on the intersection of a plane and a 3D mesh or parts of a 3D mesh defined by an SIMESH object.

Syntaxe

```
fc_siplt.sliceiso(Th,u,P)
fc_siplt.sliceiso(Th,u,P,Name,Value, ...)
```

Description

`fc_siplt.sliceiso(Th,u,P)` displays u data as isolines on the intersection of the plane defined by $P(1)x + P(2)y + P(3)z + P(4) = 0$ and all the 3-dimensional simplices elements. The data u is an 1D-array of size Th.nq or Th.nqGlobal or Th.nqParent. To compute P one can use the function PlaneCoefs of the FC-TOOLS toolbox. With this function, the array P, is obtained with P=fc_tools.graphics.PlaneCoefs(Q,V) where Q is a point in the plane and V is a vector orthogonal to the plane.

`fc_siplt.sliceiso(Th,u,P,key,value, ...)` allows additional key/value pairs to be used when displaying u. The key strings could be

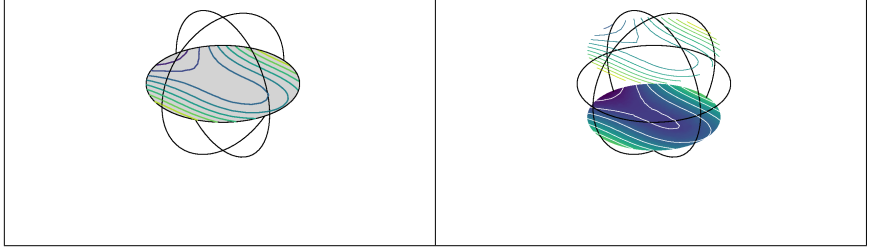
- `'labels'` : to select the labels of the elements to intersect,
- `'niso'` : to specify the number of isolines (default : 10)
- `'isorange'` : to specify the list of isovalues (default : empty)
- `'color'` : to specify one color for all isolines (default : empty)
- `'isocolorbar'` : if true display a colorbar.Default is false.
- `'format'` : to specify the format of the isovalues print in the colorbar. Default is `'%g'`.

For key strings, one could also used any options of the plot3 function.

This function accepts until 4 output arguments :

- 1st output is the colors of the isolines
- 2nd output is the isovalues of the isolines
- 3th output is the handle of the colobar iso.
- 4th output is all the handles of the isolines as an 2D-array of dimension N-by-niso, where N is the number of elementary meshes where isolines are drawn.

3D example The following example use the *.geo* file `ball8.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
meshfile=fc_oogmsh.buildmesh3d('ball8',ns*10,'verbose',0);
Th=siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y+z);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'LineWidth',1,'color','k')
hold on;axis off;axis image;
P=fc_tools.graphics.PlaneCoefs([0 0 0],[0 0 1]);
fc_siplt.slicemesh(Th,P)
fc_siplt.sliceiso(Th,u,P,'LineWidth',1.5)
figure(2)
fc_siplt.plotmesh(Th,'d',1,'LineWidth',1,'color','k')
hold on;axis off;axis image;
P2=fc_tools.graphics.PlaneCoefs([0 0 -0.5],[0 0 1]);
fc_siplt.slice(Th,u,P2)
fc_siplt.sliceiso(Th,u,P2,'color','w')
P3=fc_tools.graphics.PlaneCoefs([0 0 0.5],[0 0 1]);
fc_siplt.slice(Th,u,P3,'FaceAlpha',0.5)
fc_siplt.sliceiso(Th,u,P3,'niso',15)
```

Listing 13: 3D mesh : sliceiso function

10 PLOTQUIVER function

The method FC_SIPLT.PLOTQUIVER displays vector field datas on the mesh or parts of the mesh defined by an SIMESH object.

Syntaxe

```
fc_siplt.plotquiver(Th,V)
fc_siplt.plotquiver(Th,V,Key,Value, ...)
```

Description

fc_siplt.plotquiver(Th,V) displays vector field U on all the d-dimensional simplices elements in dimension $d = 2$ or $d = 3$. The data V is an 2D-array of size Th.nq-by-d or 2-by-Th.nq.

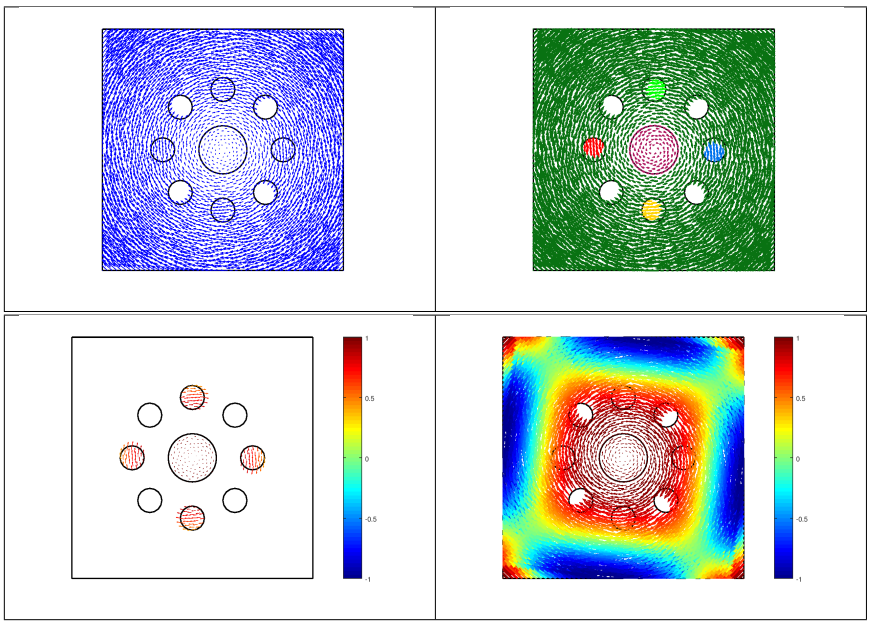
fc_siplt.plotquiver(Th,V,Key,Value, ...) specifies function options using one or more Key,Value pair arguments. Options of first level are

- `'labels'` : to select the labels of the elements to display data,
- `'freq'` : quiver frequencie, (default : 1)
- `'scale'` : quiver scale, (default : ...)
- `'colordata'` : set colors on each quiver (default : empty).

The options of second level depend on space dimension and `'colordata'` option. One can use any option of the following functions

- **quiver** function in dimension 2 with an empty `'colordata'`
- quiver3 function in dimension 3 with an empty `'colordata'`
- vfield3 function in dimension 2 or 3 with `'colordata'` set to an 1D-array of length Th.nq.

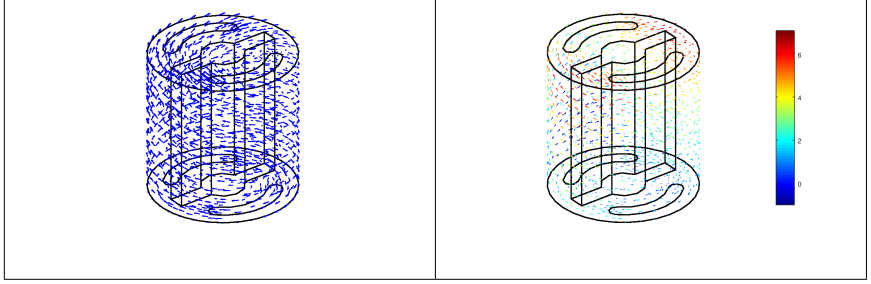
2D example The following example use the *.geo* file `condenser11.geo`.



```
geofile=fc_simesh.get_geo(2,2,'condenser11');
meshfile=fc_oogmsh.buildmesh2d(geofile,25,'verbose',0);
Th=siMesh(meshfile);
u=@(x,y) cos(pi*x.^2).*cos(pi*y.^2);
U=Th.eval(u);
w={@(x,y) y.*cos(-(x.^2+y.^2)/10),@(x,y) -x.*cos(-(x.^2+y.^2)/10)};
W=Th.eval(w);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
fc_siplt.plotquiver(Th,W)
figure(2)
fc_siplt.plotmesh(Th,'d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
fc_siplt.plotquiver(Th,W,'LineWidth',2,'merge',false)
figure(3)
fc_siplt.plotmesh(Th,'d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
fc_siplt.plotquiver(Th,W,'colordata',U,'labels',[2:2:8,20])
caxis([min(U) max(U)])
colormap('jet');colorbar
figure(4)
fc_siplt.plotmesh(Th,'d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
fc_siplt.plotquiver(Th,W,'colordata',U, 'scale',0.2)
colormap('jet');colorbar
```

Listing 14: 2D mesh : fc_siplt.plotquiver function

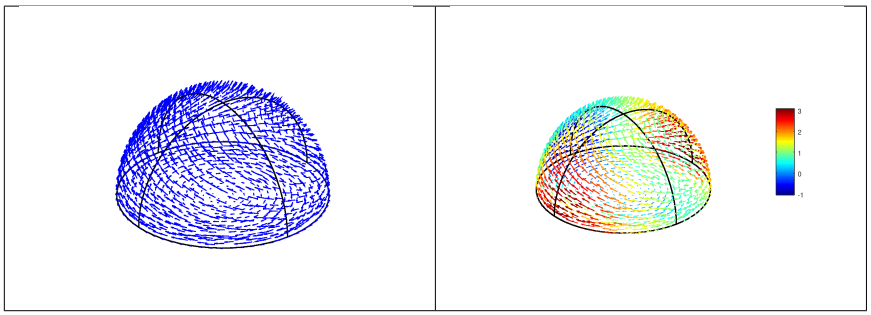
3D example The following example use the *.geo* file `cylinderkey03.geo`. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5);
Th=siMesh(meshfile);
w={@(x,y,z) y.*cos(-(x.^2+y.^2)/10),@(x,y,z) -x.*cos(-(x.^2+y.^2)/10),@(x,y,z) ...
    z/5 };
W=Th.eval(w);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k', 'LineWidth',1.5)
hold on
fc_siplt.plotquiver(Th,W,'LineWidth',1)
axis off;axis image
figure(2)
fc_siplt.plotmesh(Th,'d',1,'color','k', 'LineWidth',1.5)
hold on
fc_siplt.plotquiver(Th,W,'colordata',u, 'scale',0.05)
axis off;axis image
colormap('jet');colorbar
```

Listing 15: 3D mesh : fc_siplt.plotquiver function

3D surface example The following example use the *.geo* file `demisphere5.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D surface mesh with simplices of dimensions 1 and 2.



```
geofile=fc_simesh.get_geo(3,2,'demisphere5');
meshfile=gmsh.buildmesh3ds(geofile,ns*5,'verbose',0);
Th=siMesh(meshfile);
w={@(x,y,z) y.*cos(-(x.^2+y.^2)/10),@(x,y,z) -x.*cos(-(x.^2+y.^2)/10),@(x,y,z) z ...
    };
W=Th.eval(w);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
fc_siplt.plotquiver(Th,W,'LineWidth',1)
figure(2)
fc_siplt.plotmesh(Th,'d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
fc_siplt.plotquiver(Th,W,'colordata',u, 'scale',0.1)
colormap('jet');colorbar
```

Listing 16: 3D surface mesh : fc_siplt.plotquiver function