siplt Matlab toolbox, User's Guide*

version 0.2.0

François Cuvelier†

February 20, 2020

Abstract

This Matlab toolbox uses a **fc_simesh.siMesh** object, comming from the simesh toolbox, to display simplicial meshes or datas on simplicial meshes. Its kernel uses the graphics4mesh toolbox.

0 Contents

*LaTeX manual, revision 0.2.0.a, compiled with Matlab 2019a, and toolboxes `fc-siplt`[0.2.0], `fc-tools`[0.0.30], `fc-bench`[0.1.2], `fc-hypermesh`[1.0.3], `fc-amat`[0.1.2], `fc-meshtools`[0.1.3], `fc-graphics4mesh`[0.1.1], `fc-oogmsh`[0.2.2], `fc-simesh`[0.4.0]

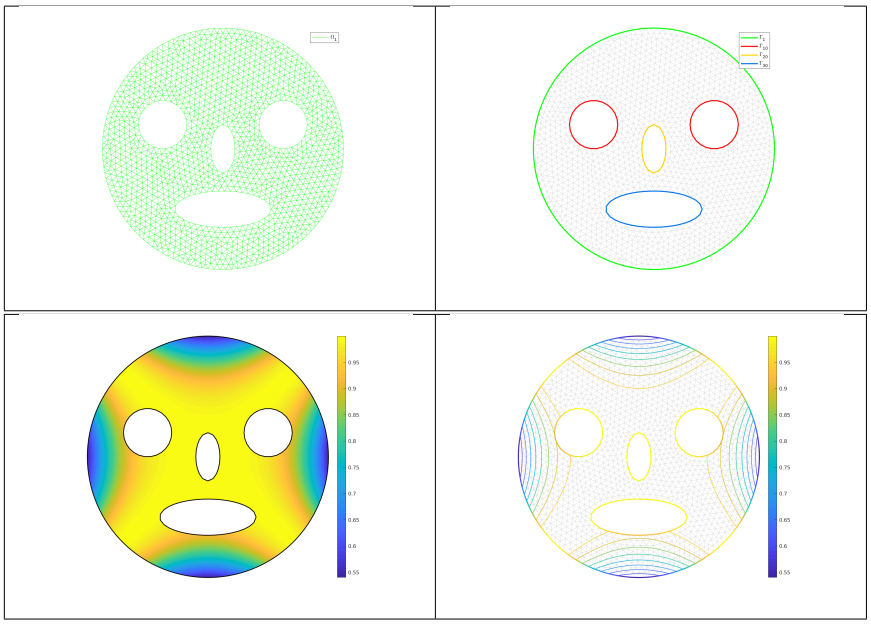
†Université Sorbonne Paris Nord, LAGA, CNRS, UMR 7539, F-93430, Villetaneuse, France, `cuvelier@math.univ-paris13.fr`.

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

1 Introduction

This **experimental** Matlab toolbox uses the graphics4mesh toolbox[1] to do some graphic representations on a **fc_simesh.siMesh** object of the simesh toolbox[2].

In Listing 1, a 2D example is provided with the 4 generated figures. For graphic representations, one can also used Th.**plotmesh**(...) instead of **fc_siplt.plotmesh**(Th,...), Th.**plot**(...) instead of **fc_siplt.plot**(Th,...) and so on.



```
close all
geofile=fc_simesh.get_geo(2,2,'sample2D01.geo');
% Using GMSH >= 4.0.0 to create mesh file
meshfile=fc_oogmsh.gmsh.buildmesh2d(geofile,200,'force',true);
% Creating siMesh object by reading the mesh file
Th=fc_simesh.siMesh(meshfile);
% Computing datas on siMesh object
u=@(x,y) cos(x.^2-y.^2);
U=Th.eval(u);
% Graphics
figure(1)
fc_siplt.plotmesh(Th,'inlegend',true)
axis image;axis off
legend()

figure(2)
fc_siplt.plotmesh(Th,'color','LightGray')
hold on
fc_siplt.plotmesh(Th,'d',1,'inlegend',true,'LineWidth',2)
axis image;axis off
legend()

figure(3)
fc_siplt.plot(Th,U,'plane',true)
colorbar
shading interp
axis image;axis off
hold on
fc_siplt.plotmesh(Th,'d',1,'LineWidth',1.5,'color','k')

figure(4)
fc_siplt.plotmesh(Th,'color','LightGray')
axis image;axis off
hold on
fc_siplt.plot(Th,U,'d',1,'LineWidth',2,'plane',true)
colorbar
fc_siplt.plotiso(Th,U,'niso',10,'LineWidth',1,'plane',true)
```

Listing 1: fc_siplt.demos.sample2D01 script with figure 1 (top left), figure 2 (top right), figure 3 (bottom left) and figure 4 (bottom right).

2 Installation

This toolbox was tested on various OS with Matlab releases:

Operating system	2017a	2017b	2018a	2018b	2019a
CentOS 7.7.1908	✓	✓	✓	✓	✓
Debian 9.11	✓	✓	✓	✓	✓
Fedora 29	✓	✓	✓	✓	✓
OpenSUSE Leap 15.0	✓	✓	✓	✓	✓
Ubuntu 18.04.3 LTS	✓	✓	✓	✓	✓
MacOS High Sierra 10.13.6	✓	✓	✓	✓	✓
MacOS Mojave 10.14.4	✓	✓	✓	✓	✓
MacOS Catalina 10.15.2	✓	✓	✓	✓	✓
Windows 10 (1909)	✓	✓	✓	✓	✓

It is not compatible with Matlab releases prior to R2015b.

2.1 Installation automatic, all in one (recommanded)

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

mfc_siplt_install.m

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

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

```
>> cd ~/Matlab/toolboxes
>> mfc_siplt_install
```

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

```
Parts of the <fc-siplt> Matlab toolbox.
Copyright (C) 2017-2020 F. Cuvelier

1- Downloading and extracting the toolboxes
2- Setting the <fc-siplt> toolbox
Write in ...
    ~/Matlab/toolboxes/fc-siplt-full/fc_siplt-0.2.0/configure_loc.m ...
3- Using toolboxes :
   ->             fc-tools : 0.0.30
   ->             fc-bench : 0.1.2
   ->         fc-hypermesh : 1.0.3
   ->              fc-amat : 0.1.2
   ->         fc-meshtools : 0.1.3
   ->     fc-graphics4mesh : 0.1.1
   ->            fc-oogmsh : 0.2.2
   ->            fc-simesh : 0.4.0
 with             fc-siplt : 0.2.0
*** Using instructions
   To use the <fc-siplt> toolbox:
   addpath('~/Matlab/toolboxes/fc-siplt-full/fc_siplt-0.2.0')
   fc_siplt.init()

   See ~/Matlab/toolboxes/mfc_siplt_set.m
```

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

```
>> addpath('~/Matlab/toolboxes/fc-siplt-full/mfc-siplt-0.2.0')
>> fc_siplt.init()
```

If it's the first time the `fc_siplt.init()` function is used, then its output is

```
Try to use default parameters!
 Use fc_tools.configure to configure.
Write in ...
    ~/Matlab/toolboxes/fc-siplt-full/fc_tools-0.0.30/configure_loc.m ...
Try to use default parameters!
 Use fc_bench.configure to configure.
Write in ...
    ~/Matlab/toolboxes/fc-siplt-full/fc_bench-0.1.2/configure_loc.m ...
Try to use default parameters!
 Use fc_hypermesh.configure to configure.
Write in ...
    ~/Matlab/toolboxes/fc-siplt-full/fc_hypermesh-1.0.3/configure_loc.m ...
    ...
Try to use default parameters!
 Use fc_amat.configure to configure.
Write in ...
    ~/Matlab/toolboxes/fc-siplt-full/fc_amat-0.1.2/configure_loc.m ...
Try to use default parameters!
 Use fc_meshtools.configure to configure.
Write in ...
    ~/Matlab/toolboxes/fc-siplt-full/fc_meshtools-0.1.3/configure_loc.m ...
    ...
Try to use default parameters!
 Use fc_graphics4mesh.configure to configure.
Write in ...
    ~/Matlab/toolboxes/fc-siplt-full/fc_graphics4mesh-0.1.1/configure_loc.m ...
    ...
Try to use default parameters!
 Use fc_oogmsh.configure to configure.
Write in ...
    ~/Matlab/toolboxes/fc-siplt-full/fc_oogmsh-0.2.2/configure_loc.m ...
Configured to use gmsh 4.5.1 with default MSH file format version 4.1
Try to use default parameters!
 Use fc_simesh.configure to configure.
Write in ...
    ~/Matlab/toolboxes/fc-siplt-full/fc_simesh-0.4.0/configure_loc.m ...
Using fc_siplt[0.2.0] with fc_tools[0.0.30], fc_bench[0.1.2], ...
    fc_hypermesh[1.0.3],
      fc_amat[0.1.2], fc_meshtools[0.1.3], fc_graphics4mesh[0.1.1], ...
          fc_oogmsh[0.2.2], fc_simesh[0.4.0].
```

Otherwise, the output of the `fc_siplt.init()` function is

```
Configured to use gmsh 4.5.1 with default MSH file format version 4.1
Using fc_siplt[0.2.0] with fc_tools[0.0.30], fc_bench[0.1.2], ...
    fc_hypermesh[1.0.3],
      fc_amat[0.1.2], fc_meshtools[0.1.3], fc_graphics4mesh[0.1.1], ...
          fc_oogmsh[0.2.2], fc_simesh[0.4.0].
```

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

`~/Matlab/toolboxes/fc-siplt-full`

3 fc_siplt.plotmesh function

The **fc_siplt.plotmesh** function displays the mesh or parts of the mesh defined by an **fc_simesh.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 of Th, a **fc_simesh.siMesh** object.

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**)
- 'cutPlane' : cut mesh by n plans given by n-by-4 array P where the equation of the i-th cut plane 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.

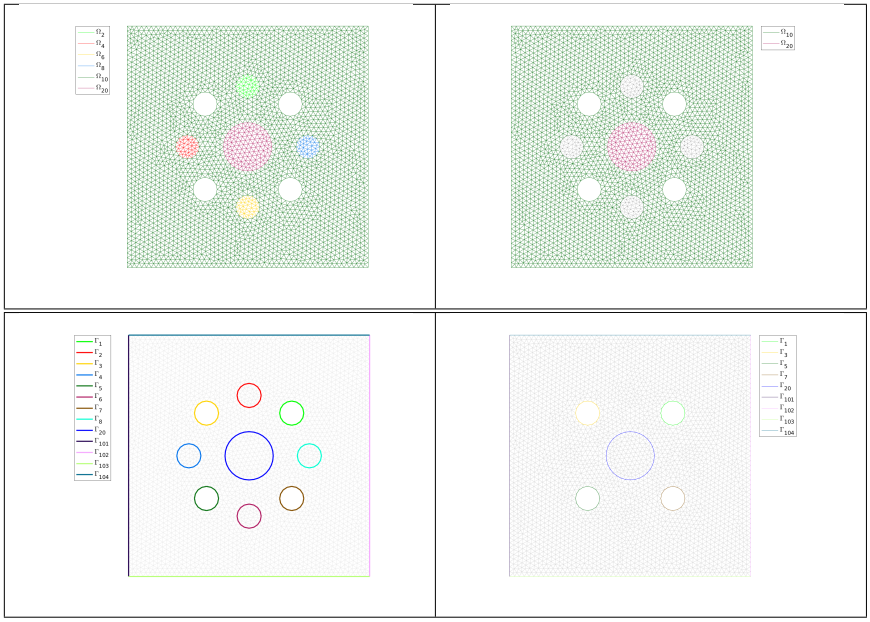
- In dimension 3,
  - if $d == 3$, **patch** function is used,
  - if $d == 2$, **trimesh** function is used,
  - if $d == 1$, **plot3** function is used,
  - if $d == 0$, **plot3** function is used,
- In dimension 2,
  - if $d == 2$, **trimesh** function is used,
  - if $d == 1$, **plot** function is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

3.1 2D example

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

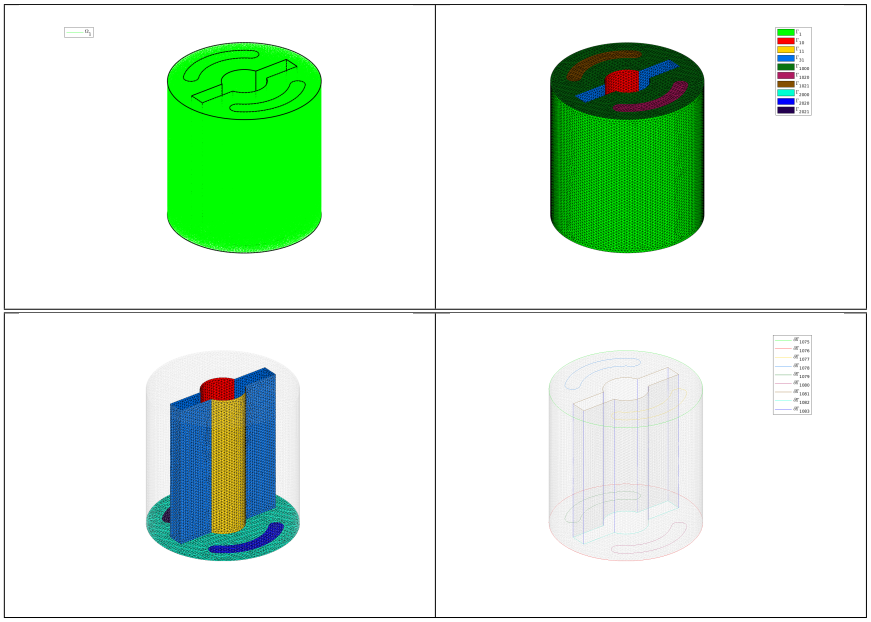


```
meshfile=fc_oogmsh.gmsh.buildmesh2d('condenser11',25,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
figure(1)
fc_siplt.plotmesh(Th,'inlegend',true)
axis off;axis image;
set(legend('show'),'Location','NorthWestOutside','fontsize',12)
figure(2)
fc_siplt.plotmesh(Th,'labels',[10,20],'inlegend',true)
hold 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 2: 2D mesh : **fc_siplt.plotmesh** function

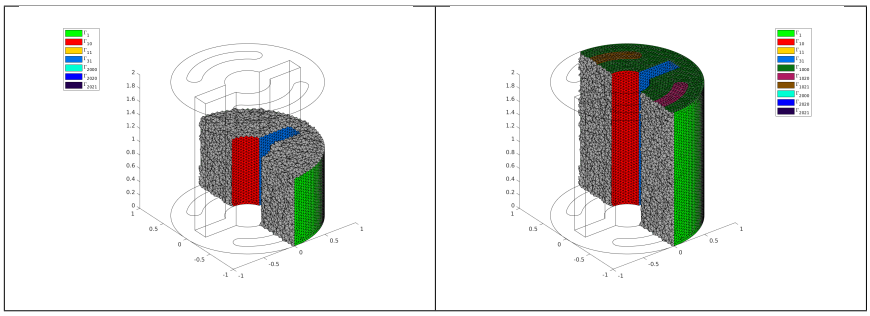
3.2 3D example

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=fc_simesh.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 3: 3D plot mesh

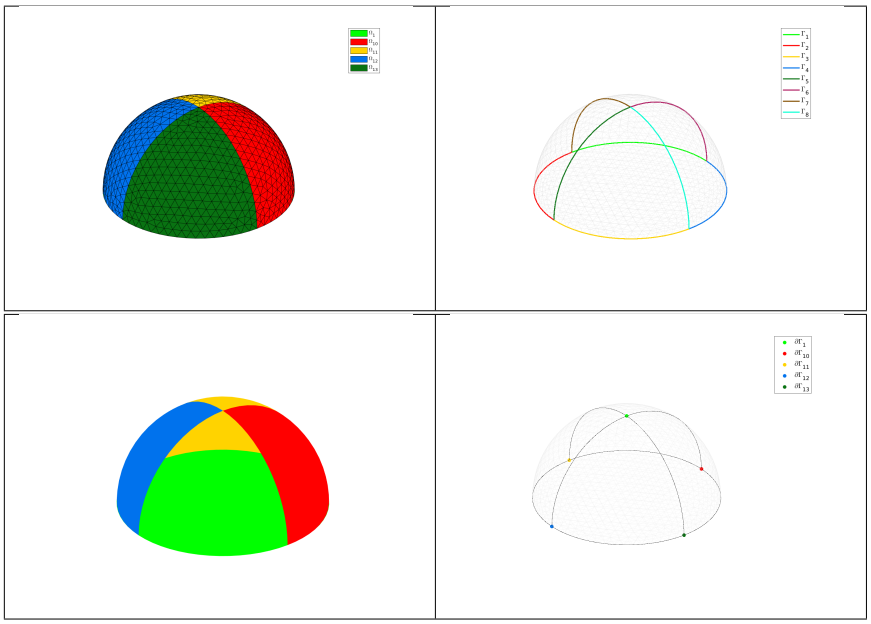


```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5,'verbose',0);
figure(5)
P=[fc_tools.graphics.PlaneCoefs([0 0 1], [0 0 1]); ...
    fc_tools.graphics.PlaneCoefs([0 0 1],[-1 0 0])];
fc_siplt.plotmesh(Th,'d',1,'color','black')
hold on;axis image;view(-75,36)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

Listing 4: 3D mesh : **fc_siplt.plotmesh** function

3.3 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=fc_simesh.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')
axis off;axis equal
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 5: 3D surface mesh : fc_siplt.plotmesh function

4 fc_siplt.plot function

The **fc_siplt.plot** function displays scalar datas on the mesh or parts of the mesh defined by an **fc_simesh.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 of Th, a **fc_simesh.siMesh** object. 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

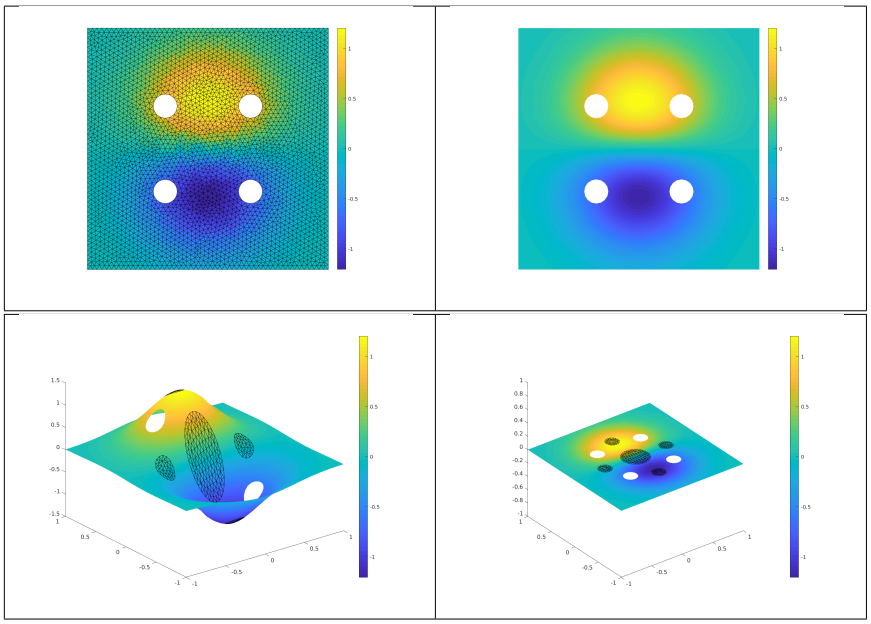
- 'd' : to specify the dimension of the simplices elements (default : Th.d)
- 'labels' : to select the labels of the elements to display data,
- 'plane' : if true, made a 2D representation in the xy-plane, otherwise made a 3D representation with z-value set to u (default : **false**)

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 'plane' option is true, **patch** function is used, otherwise it's **trisurf** function,
  - for $d == 1$, **patch** function is used.
- In dimension 1 and $d == 1$, **plot** function is used

4.1 2D example

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

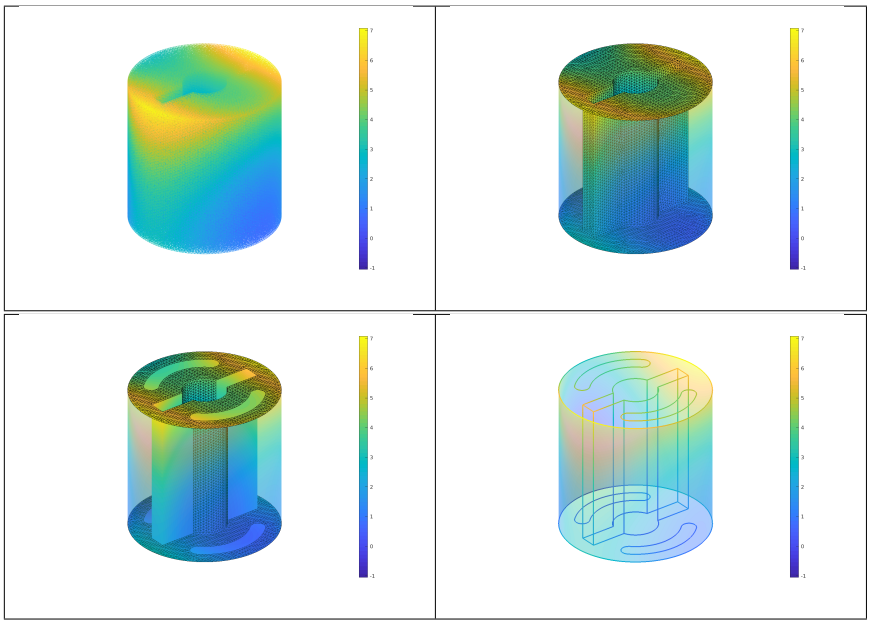


```
geofile=fc_simesh.get_geo(2,2,'condenser11');
meshfile=fc_oogmsh.gmsh.buildmesh2d(geofile,25,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y) 5*exp(-3*(x.^2+y.^2)).*cos(x).*sin(y));
figure(1)
fc_siplt.plot(Th,u)
axis off;axis image;colorbar
figure(2)
fc_siplt.plot(Th,u)
axis off;axis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

Listing 6: 2D mesh : fc_siplt.plot function

4.2 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.

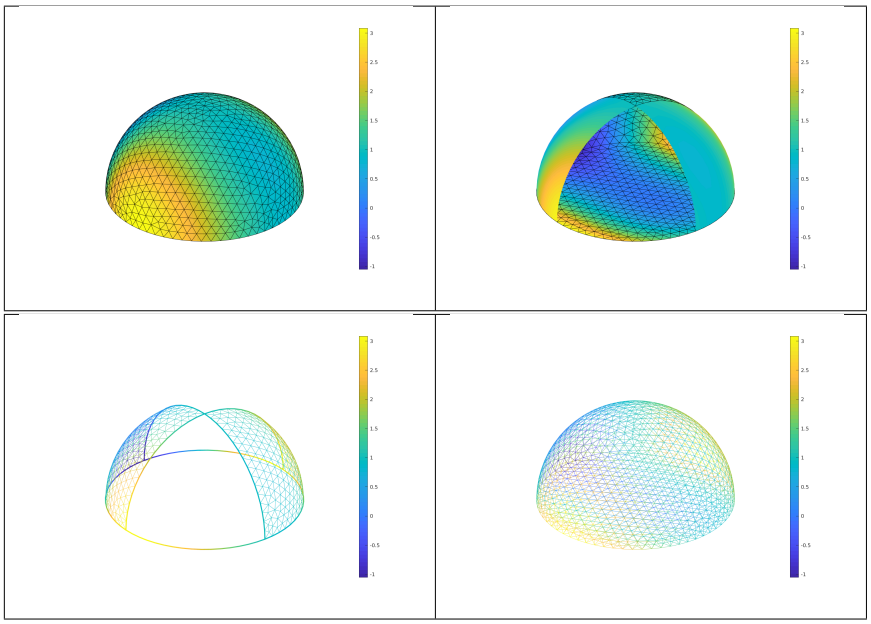


```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5);
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plot(Th,u);
axis off;axis image;colorbar
figure(2)
fc_siplt.plot(Th,u,'d',2,'labels',[10,11,31,1000,1020,1021,2000,2020,2021])
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 7: 3D mesh : fc_siplt.plot function

4.3 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=fc_simesh.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 8: 3D surface mesh : fc_siplt.plot function

5 fc_siplt.plotiso function

The **fc_siplt.plotiso** function displays isolines from datas on the mesh or parts of the mesh defined by an **fc_simesh.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 of Th, a **fc_simesh.siMesh** object.. 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,
- 'plane' : if true, isolines are in the xy-plane, otherwise isolines are in 3D with z-value set to u (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

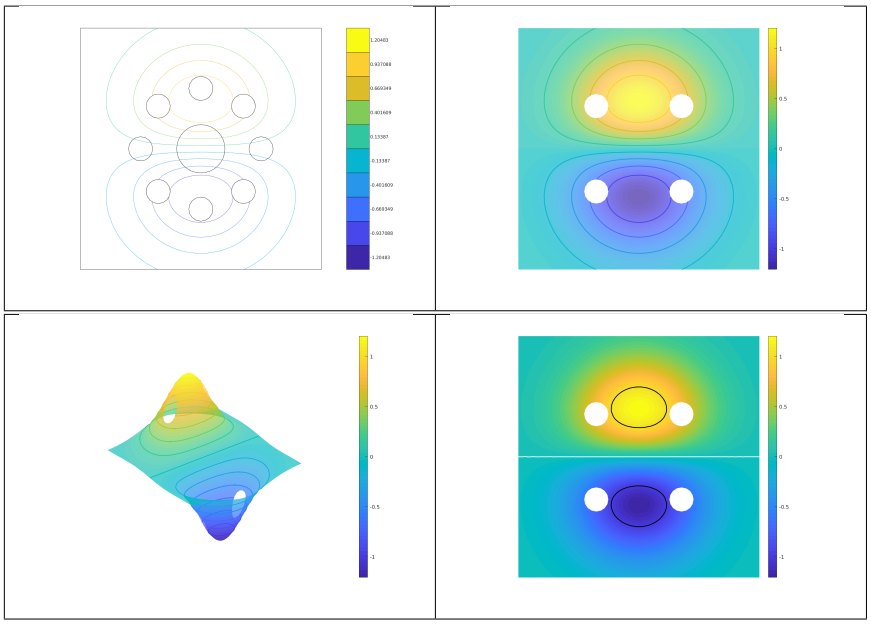
- **plot3** function in dimension 3 or in dimension 2 with 'plane' option set to **false**
- **plot** function in 2 with 'plane' option set to **true**

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 2-simplex elementary meshes where isolines are drawn.

5.1 2D example

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

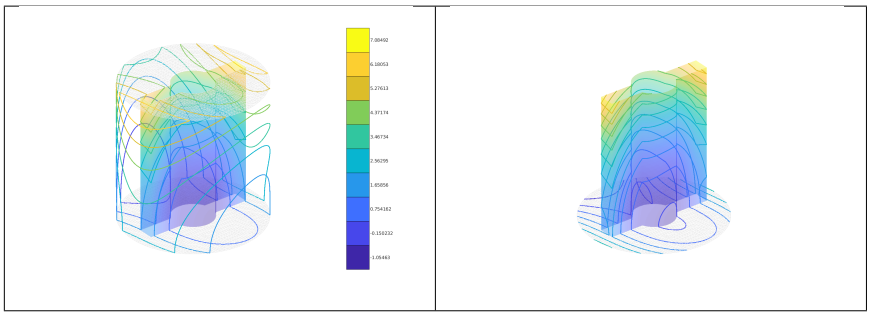


```
meshfile=fc_oogmsh.gmsh.buildmesh2d('condenser11',25,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y) 5*exp(-3*(x.^2+y.^2)).*cos(x).*sin(y));
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k')
hold on;axis off;axis image;
fc_siplt.plotiso(Th,u,'isocolorbar',true)
figure(2)
fc_siplt.plot(Th,u,'plane',true,'FaceAlpha',0.7)
hold on;axis off;axis image;shading interp;
fc_siplt.plotiso(Th,u,'plane',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,'plane',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','plane',true)
axis off;axis image;colorbar
```

Listing 9: 2D mesh : **fc_siplt.plotiso** function

5.2 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.

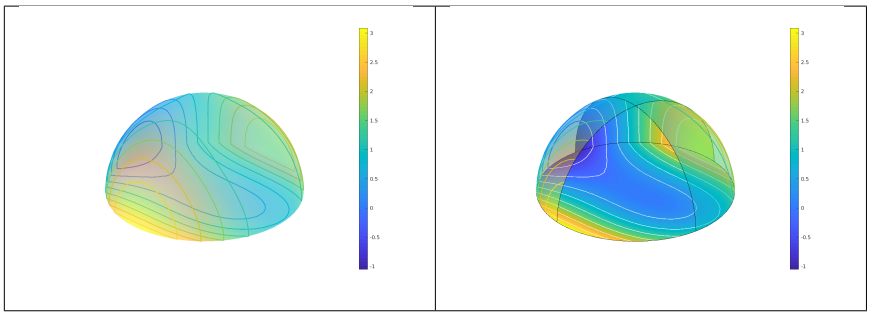


```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.gmsh.buildmesh3d(geofile,ns*5,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plot(Th,u,'d',2,'labels',[10,11,31],'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
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 10: 3D mesh : fc_siplt.plotiso function

5.3 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.gmsh.buildmesh3ds('demisphere5',ns*5,'verbose',0);
Th=fc_simesh.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 11: 3D surface mesh : fc_siplt.plotiso function

6 fc_siplt.slicemesh function

The **fc_siplt.slicemesh** function displays intersection of a plane and a 3D mesh or parts of a 3D mesh defined by an **fc_simesh.siMesh** object.

Syntaxe

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

Description

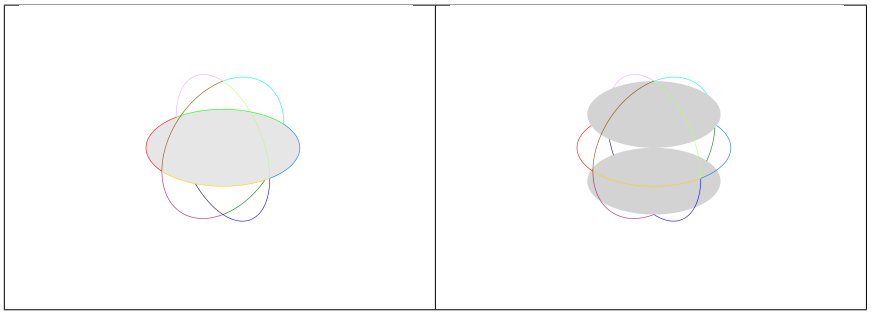
fc_siplt.slicemesh(Th,P) displays intersection of the plane defined by $P(1)x+P(2)y+P(3)z+P(4)=0$ and all the 3-dimensional simplices elements of Th, a **fc_simesh.siMesh** object. To compute P one can use the function fc_tools.graphics.PlaneCoefs of the fc tools toolbox. With this function, the array 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 it.

fc_siplt.slicemesh(Th,P,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 : 'lightgrey', rgb =[0.9,0.9,0.9])
- 'labels' : to select the labels of the elements to intersect,

6.1 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=fc_simesh.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 12: 3D mesh : fc_siplt.slicemesh function

7 fc_siplt.slice function

The method **fc_siplt.slice** function displays datas on the intersection of a plane and a 3D mesh or parts of a 3D mesh defined by an **fc_simesh.siMesh** object.

Syntaxe

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

Description

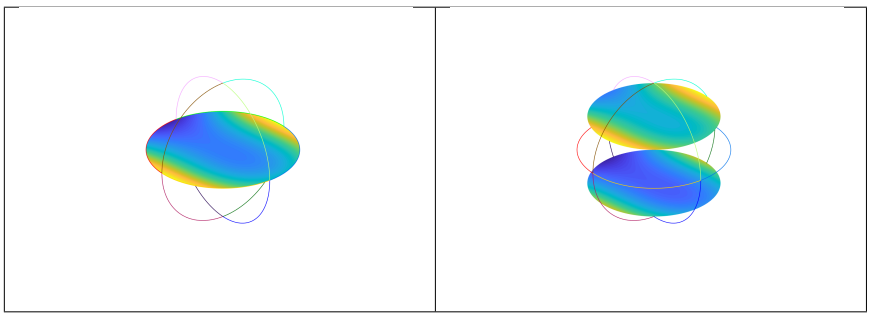
fc_siplt.slice(Th,u,P) displays u data on the intersection of the plane defined by $P(1)x + P(2)y + P(3)z + P(4) = 0$ and all the 3-dimensional simplices elements of Th, a **fc_simesh.siMesh** object. The data u is an 1D-array of size Th.nq or Th.nqGlobal or Th.nqParent. To compute P one can use the function fc_tools.graphics.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 it.

fc_siplt.slice(Th,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,

7.1 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=fc_simesh.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 13: 3D mesh :fc_siplt.slice function

8 fc_siplt.sliceiso function

The **fc_siplt.sliceiso** function displays isolines of data on the intersection of a plane and a 3D mesh or parts of a 3D mesh defined by an **fc_simesh.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 of Th, a **fc_simesh.siMesh** object. 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 fc_tools.graphics.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'`.

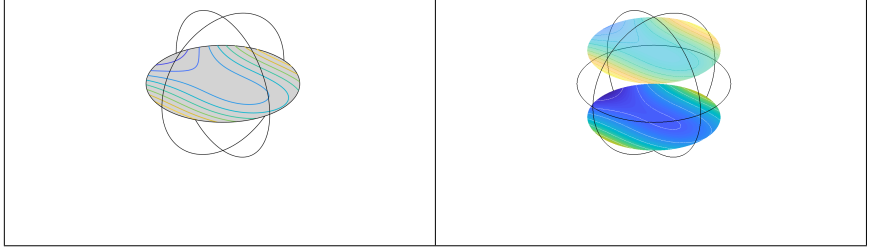
For key strings, one could also used any options of the plot3 function.

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.

8.1 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=fc_simesh.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 14: 3D mesh :fc_siplt.sliceiso function

9 fc_siplt.plotquiver function

The **fc_siplt.plotquiver** function displays vector field datas on the mesh or parts of the mesh defined by an **fc_simesh.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

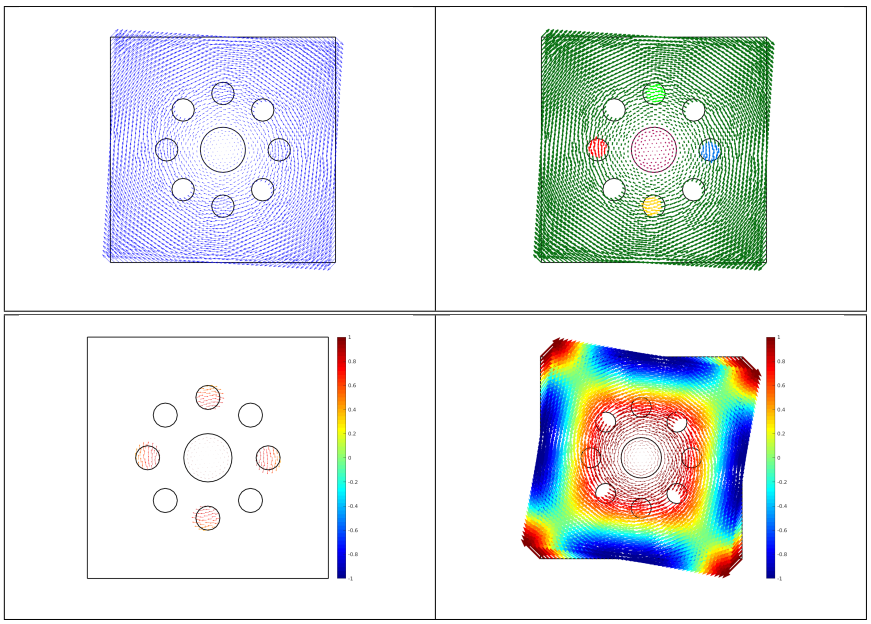
- 'labels' : to select the labels of the elements to display data,
- 'freq' : quiver frequencie, (default : 1)
- 'scale' : quiver scale, (default : ...)
- 'colordata' : set colors on each quiver (default : empty).

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.

9.1 2D example

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

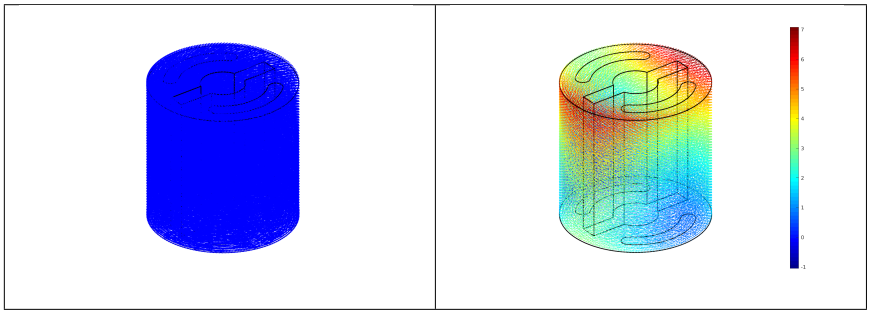


```
geofile=fc_simesh.get_geo(2,2,'condenser11');
meshfile=fc_oogmsh.buildmesh2d(geofile,25,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=@(x,y) cos(pi*x.^2).*cos(pi*y.^2);
U=Th.eval(u);
w={@(x,y) y.*cos(-(x.^2+y.^2)/10),@(x,y) -x.*cos(-(x.^2+y.^2)/10)};
W=Th.eval(w);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k', '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 15: 2D mesh : fc_siplt.plotquiver function

9.2 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.

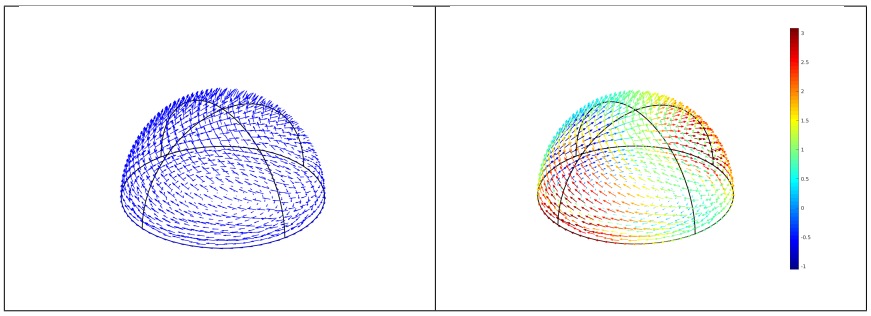


```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5);
Th=fc_simesh.siMesh(meshfile);
w={@(x,y,z) y.*cos(-(x.^2+y.^2)/10),@(x,y,z) -x.*cos(-(x.^2+y.^2)/10),@(x,y,z) ...
    z/5 };
W=Th.eval(w);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k', '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 16: 3D mesh : `fc_siplt.plotquiver` function

9.3 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.gmsh.buildmesh3ds(geofile,ns*5,'verbose',0);
Th=fc_simesh.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 17: 3D surface mesh : **fc_siplt.plotquiver** function

Appendices

A Listings

B References

[1] F. Cuvelier. fc_graphics4mesh: a Matlab toolbox for displaying simplices meshes or datas on simplices meshes. `http://www.math.univ-paris13.fr/~cuvelier/software/`, 2017. User's Guide.

[2] F. Cuvelier. fc_simesh: an object-oriented Matlab toolbox for using simplices meshes generated from gmsh (in dimension 2 or 3) or an hypercube triangulation (in any dimension). `http://www.math.univ-paris13.fr/~cuvelier/software/`, 2017. User's Guide.

Informations for git maintainers of the fc siplt Matlab toolbox

git informations on the toolboxes used to build this manual

```
-------------------------------------------------
  name : fc-siplt
   tag : 0.2.0
commit : e9cac51a2239a52ce34125ddcd2f1de4377dda5f
  date : 2020-02-20
  time : 09-50-32
status : 0
-------------------------------------------------
  name : fc-tools
   tag : 0.0.30
commit : 42054f06c3f484c5fc7d0a0cb425d727b50e8994
  date : 2020-02-18
  time : 06-46-53
status : 0
-------------------------------------------------
  name : fc-bench
   tag : 0.1.2
commit : 666dc60d1277f5fa9c99dee4ae1c33270f22c57d
  date : 2020-02-16
  time : 06-38-46
status : 0
-------------------------------------------------
  name : fc-hypermesh
   tag : 1.0.3
commit : c520b34cfd7eb0dbf9e4ecd459fd7162db73cc58
  date : 2020-02-16
  time : 08-34-19
status : 0
-------------------------------------------------
  name : fc-amat
   tag : 0.1.2
commit : 957340f6e71d805dbd8b9d04c434b24fd3f92591
  date : 2020-02-16
  time : 06-39-42
status : 0
-------------------------------------------------
  name : fc-meshtools
   tag : 0.1.3
commit : cdbc41bc98af4e4faccc1746024aced1f21aae53
  date : 2020-02-17
  time : 10-52-56
status : 0
-------------------------------------------------
  name : fc-graphics4mesh
   tag : 0.1.1
commit : d8e283fc3e1ce5fac855064062e196750a0a0cd4
  date : 2020-02-19
  time : 08-50-23
status : 0
-------------------------------------------------
  name : fc-oogmsh
   tag : 0.2.2
commit : 90fc9826fea6eb8dd66afc3a58677564c8f7f442
  date : 2020-02-17
  time : 13-31-05
status : 0
-------------------------------------------------
  name : fc-simesh
   tag : 0.4.0
commit : 9e234702263df3e1dc19cfff57bbdf5248859725
  date : 2020-02-20
  time : 10-01-30
status : 0
-------------------------------------------------
```

git informations on the LaTeX package used to build this manual

```
--------------------------------------------------
   name : fctools
    tag :
 commit : 57968c4a96c2593cccc9da9efd3e52b2ff012cb5
   date : 2020-02-07
   time : 06:41:09
 status : 1
--------------------------------------------------
```

Using the remote configuration repository:

url	`ssh://lagagit/MCS/Cuvelier/Matlab/fc-config`
commit	`b837af396f81fb38264a3affb2f2d5c779818b6c`