siplt Matlab toolbox, User's Guide*

version 0.2.5

François Cuvelier†

January 5, 2023

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 0.2.5.a, compiled with Matlab 2022a, and toolboxes `fc-siplt`[0.2.5], `fc-tools`[0.0.34], `fc-bench`[0.1.3], `fc-hypermesh`[1.0.4], `fc-amat`[0.1.3], `fc-meshtools`[0.1.4], `fc-graphics4mesh`[0.1.5], `fc-oogmsh`[0.2.4], `fc-simesh`[0.4.5]

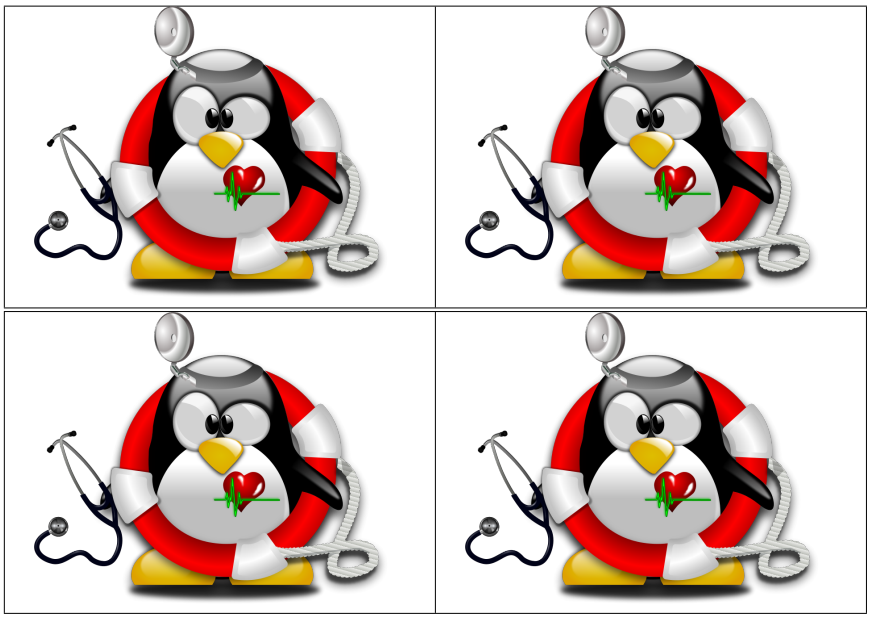
†Université Sorbonne Paris Nord, LAGA, CNRS, UMR 7539, F-93430, Villetaneuse, France, `cuvelier@math.univ-paris13.fr`.

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 only tested on Ubuntu 22.04.1 with Matlab R2022a.

One just has 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* toolbox and all the required required toolboxes in the current directory.

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

```
>> cd ~/Matlab
>> 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) 2017-2022 F. Cuvelier

1- Downloading and extracting the toolboxes
2- Setting the <fc-siplt> toolbox
Write in ~/Matlab/fc-siplt-full/fc_siplt-0.2.5/configure_loc.m ...
3- Using toolboxes :
   ->              fc-tools : 0.0.34
   ->              fc-bench : 0.1.3
   ->         fc-hypermesh : 1.0.4
   ->               fc-amat : 0.1.3
   ->         fc-meshtools : 0.1.4
   ->     fc-graphics4mesh : 0.1.5
   ->             fc-oogmsh : 0.2.4
   ->             fc-simesh : 0.4.5
 with              fc-siplt : 0.2.5
*** Using instructions
   To use the <fc-siplt> toolbox:
   addpath('~/Matlab/fc-siplt-full/fc_siplt-0.2.5')
   fc_siplt.init()

   See ~/Matlab/mfc_siplt_set.m
```

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

```
>> addpath('~/Matlab/fc-siplt-full/mfc-siplt-0.2.5')
>> 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/fc-siplt-full/fc_tools-0.0.34/configure_loc.m ...
Try to use default parameters!
 Use fc_bench.configure to configure.
Write in ~/Matlab/fc-siplt-full/fc_bench-0.1.3/configure_loc.m ...
Try to use default parameters!
 Use fc_hypermesh.configure to configure.
Write in ~/Matlab/fc-siplt-full/fc_hypermesh-1.0.4/configure_loc.m ...
Try to use default parameters!
 Use fc_amat.configure to configure.
Write in ~/Matlab/fc-siplt-full/fc_amat-0.1.3/configure_loc.m ...
Try to use default parameters!
 Use fc_meshtools.configure to configure.
Write in ~/Matlab/fc-siplt-full/fc_meshtools-0.1.4/configure_loc.m ...
Try to use default parameters!
 Use fc_graphics4mesh.configure to configure.
Write in ~/Matlab/fc-siplt-full/fc_graphics4mesh-0.1.5/configure_loc.m ...
    ...
Try to use default parameters!
 Use fc_oogmsh.configure to configure.
Write in ~/Matlab/fc-siplt-full/fc_oogmsh-0.2.4/configure_loc.m ...
Try to use default parameters!
 Use fc_simesh.configure to configure.
Write in ~/Matlab/fc-siplt-full/fc_simesh-0.4.5/configure_loc.m ...
Using fc_siplt[0.2.5] with fc_tools[0.0.34], fc_bench[0.1.3], ...
    fc_hypermesh[1.0.4],
                          fc_amat[0.1.3], fc_meshtools[0.1.4], ...
                              fc_graphics4mesh[0.1.5],
                          fc_oogmsh[0.2.4], fc_simesh[0.4.5].
[fc-oogmsh] Configured to use gmsh 4.11.0 with default MSH file format ...
    version 4.1
```

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

```
Using fc_siplt[0.2.5] with fc_tools[0.0.34], fc_bench[0.1.3], ...
    fc_hypermesh[1.0.4],
                          fc_amat[0.1.3], fc_meshtools[0.1.4], ...
                              fc_graphics4mesh[0.1.5],
                          fc_oogmsh[0.2.4], fc_simesh[0.4.5].
[fc-oogmsh] Configured to use gmsh 4.11.0 with default MSH file format ...
    version 4.1
```

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

`~/Matlab/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

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

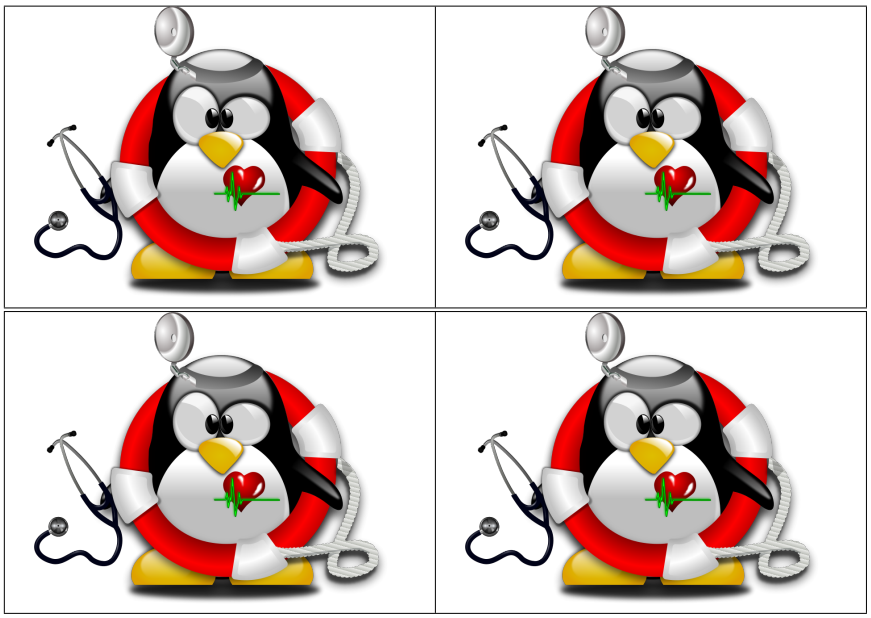
  The normal vector $P(i,1:3)$ pointed to the part of the mesh not displayed. (only for simplices in dimension 3) default : [] (no cut).

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 ....

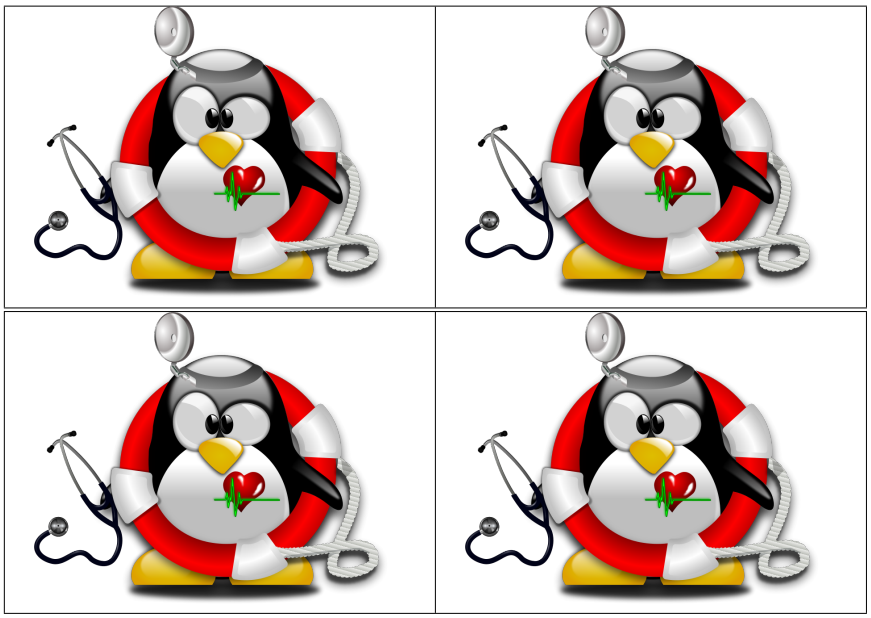


```
meshfile=fc_oogmsh.gmsh.buildmesh2d('condenser11',25,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
figure(1)
fc_siplt.plotmesh(Th,'inlegend',true)
axis off;axis image;
set(legend('show'),'Location','NorthWestOutside','fontsize',12)
figure(2)
fc_siplt.plotmesh(Th,'labels',[10,20],'inlegend',true)
hold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

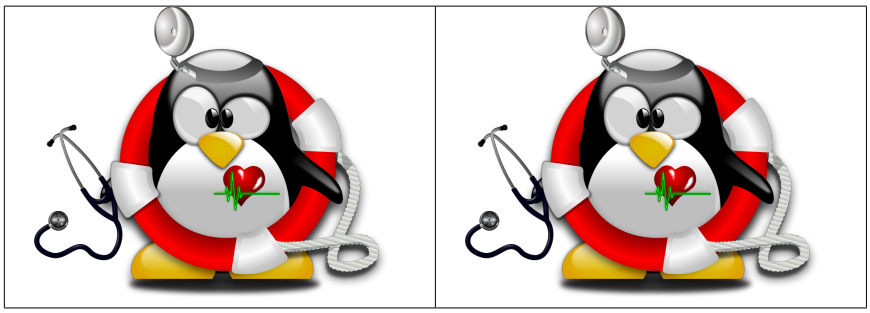
3.2 3D example

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

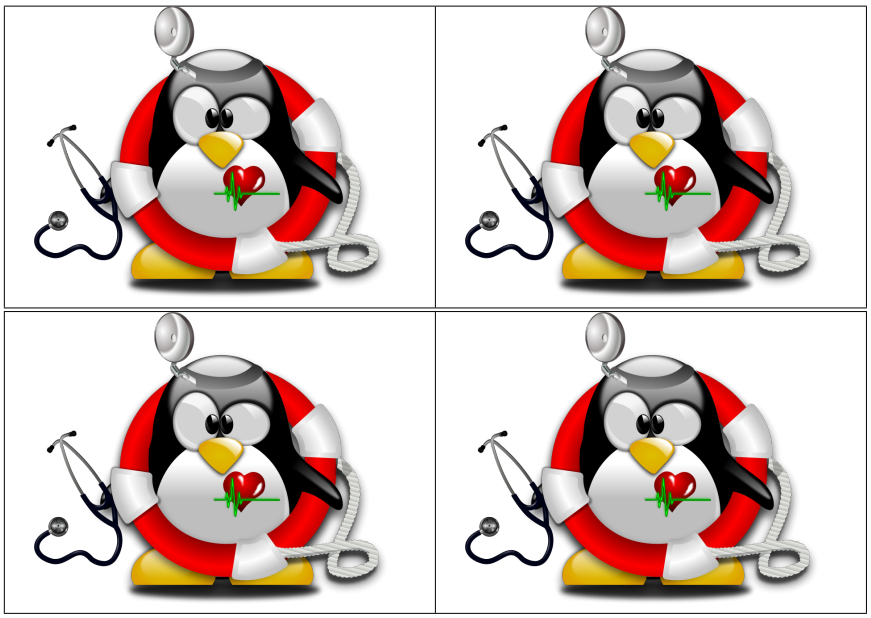


```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5,'verbose',0);
figure(5)
P=[fc_tools.graphics.PlaneCoefs([0 0 1], [0 0 1]); ...
   fc_tools.graphics.PlaneCoefs([0 0 1],[-1 0 0])];
fc_siplt.plotmesh(Th,'d',1,'color','black')
hold on;axis image;view(-75,36)
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

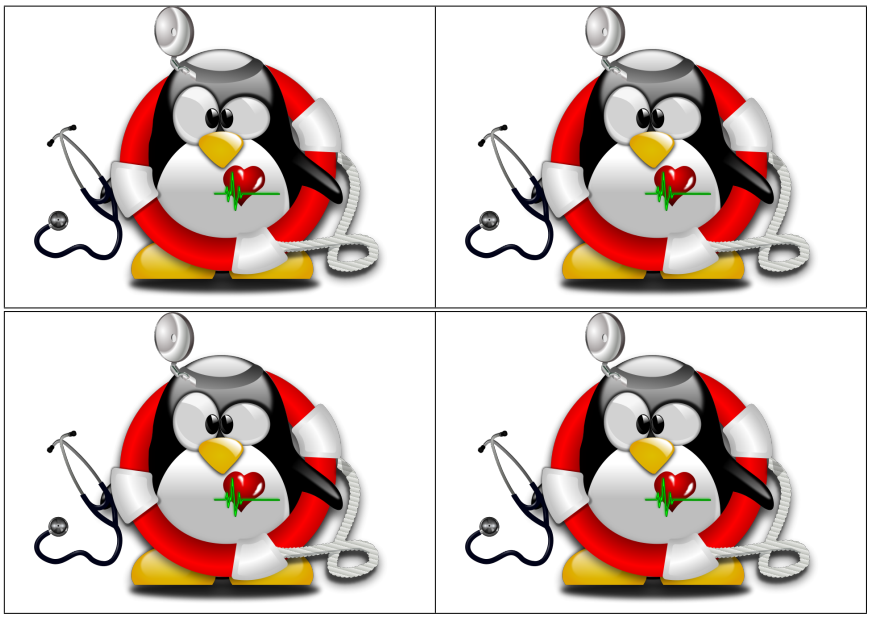
- 'd' : to specify the dimension of the simplices elements (default : Th.d)
- 'labels' : to select the labels of the elements to display data,
- 'plane' : if true, made a 2D representation in the xy-plane, otherwise made a 3D representation with z-value set to u (default : **false**)

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.

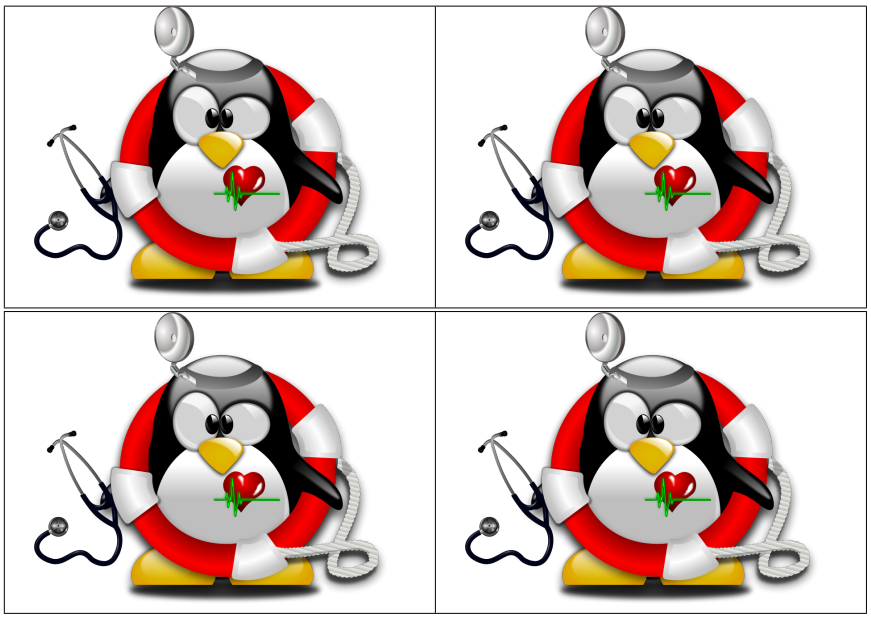


```
geofile=fc_simesh.get_geo(2,2,'condenser11');
meshfile=fc_oogmsh.gmsh.buildmesh2d(geofile,25,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y) 5*exp(-3*(x.^2+y.^2)).*cos(x).*sin(y));
figure(1)
fc_siplt.plot(Th,u)
axis off;axis image;colorbar
figure(2)
fc_siplt.plot(Th,u)
axis off;axis 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.

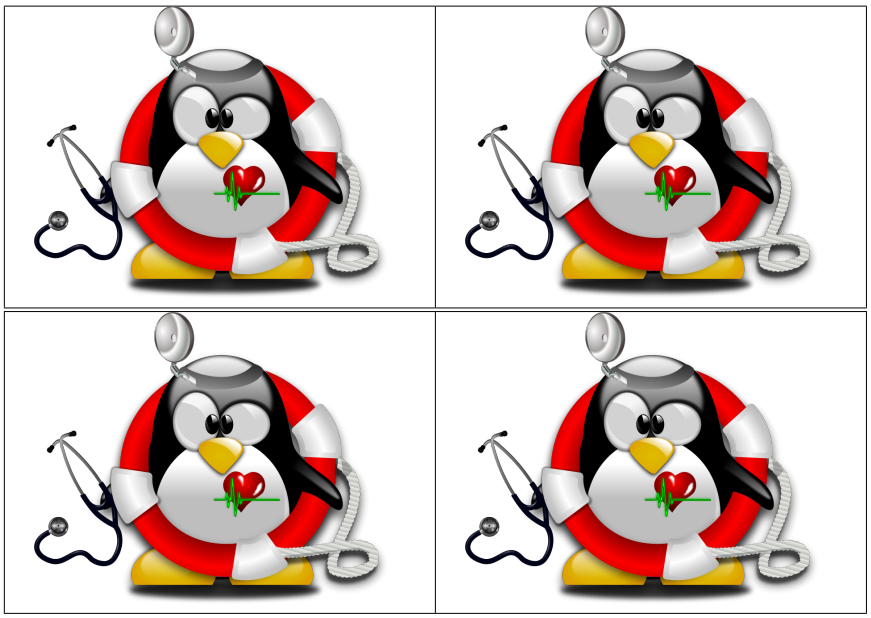


```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5);
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plot(Th,u);
axis off;axis image;colorbar
figure(2)
fc_siplt.plot(Th,u,'d',2,'labels',[10,11,31,1000,1020,1021,2000,2020,2021])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

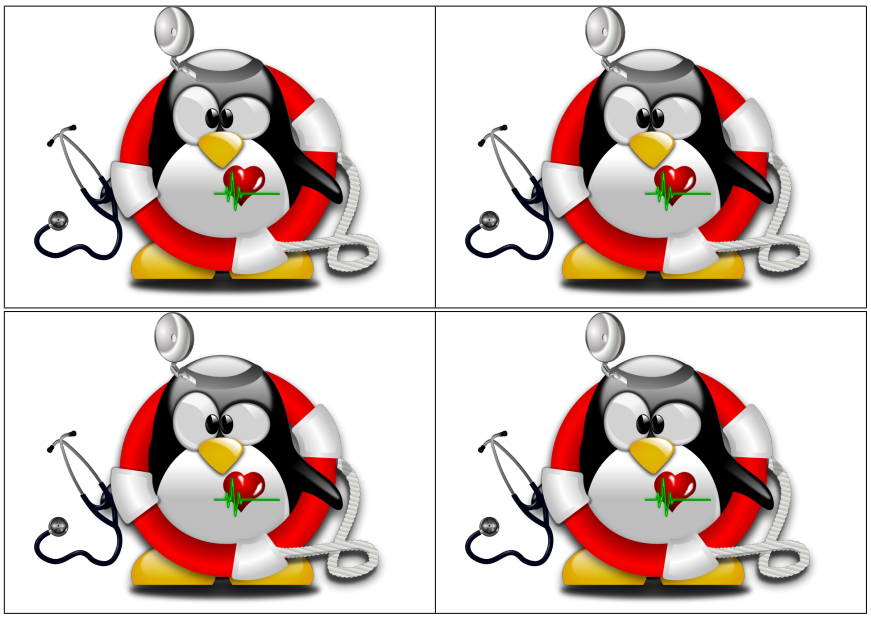
- **plot3** function in dimension 3 or in dimension 2 with 'plane' option set to **false**
- **plot** function in 2 with 'plane' option set to **true**

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.

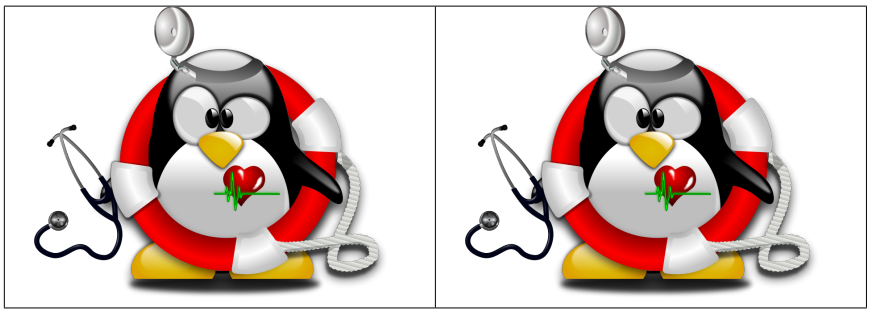


```
meshfile=fc_oogmsh.gmsh.buildmesh2d('condenser11',25,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y) 5*exp(-3*(x.^2+y.^2)).*cos(x).*sin(y));
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k')
hold on;axis off;axis image;
fc_siplt.plotiso(Th,u,'isocolorbar',true)
figure(2)
fc_siplt.plot(Th,u,'plane',true,'FaceAlpha',0.7)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

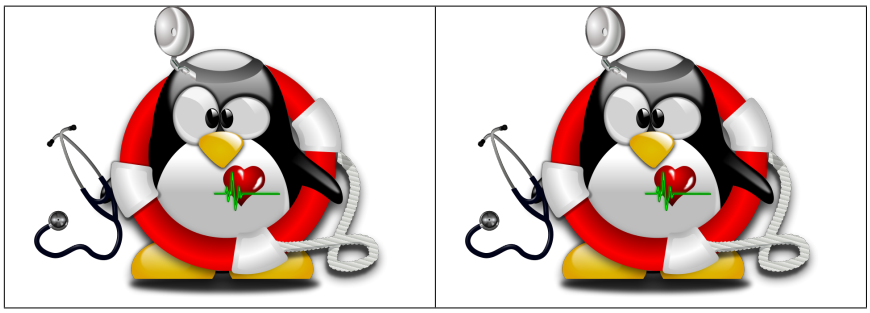


```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.gmsh.buildmesh3d(geofile,ns*5,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plot(Th,u,'d',2,'labels',[10,11,31],'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)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

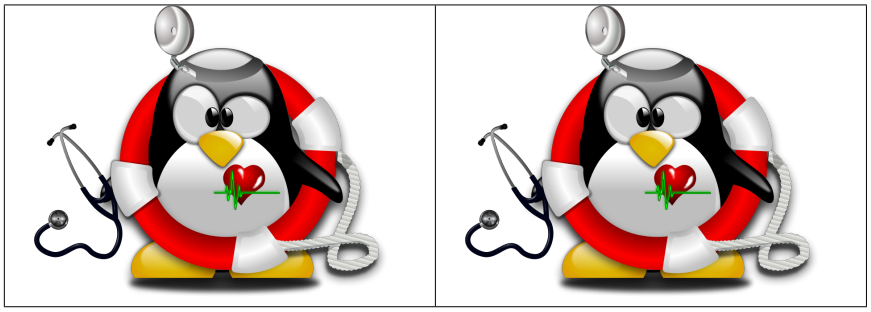
fc_siplt.slicemesh(Th,P) displays intersection of the plane defined by $P(1)x+P(2)y+P(3)z+P(4)=0$ and all the 3-dimensional simplices elements of Th, a **fc_simesh.siMesh** object. To compute P one can use the function fc_tools.graphics.PlaneCoefs of the fctools toolbox. With this function, the array 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*5,'verbose',0,'force',true);
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

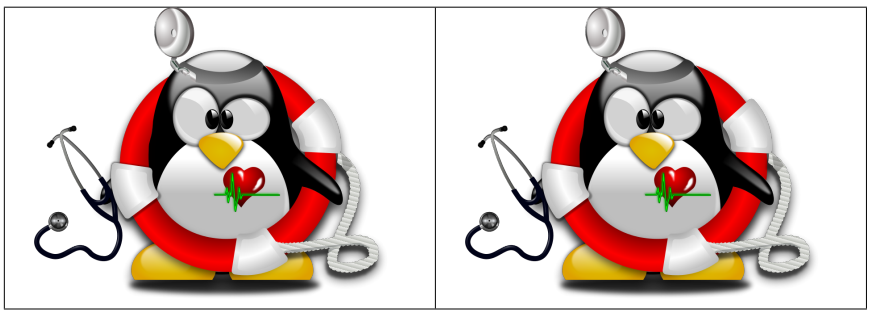
fc_siplt.slice(Th,u,P) displays u data on the intersection of the plane defined by $P(1)x + P(2)y + P(3)z + P(4) = 0$ and all the 3-dimensional simplices elements of Th, a **fc_simesh.siMesh** object. The data u is an 1D-array of size Th.nq or Th.nqGlobal or Th.nqParent. To compute P one can use the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

fctools 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'`.

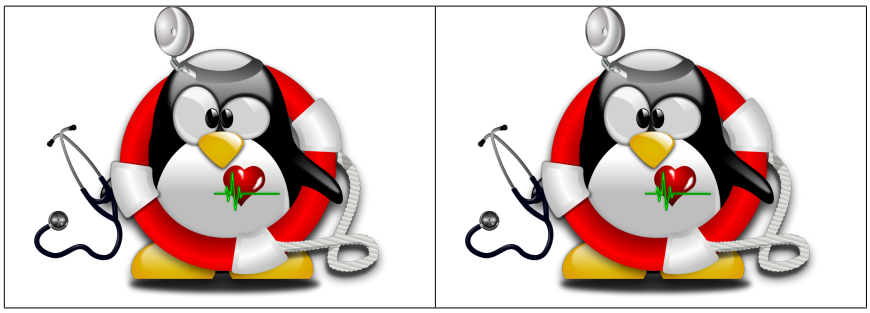
For key strings, one could also used any options of the plot3 function.

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

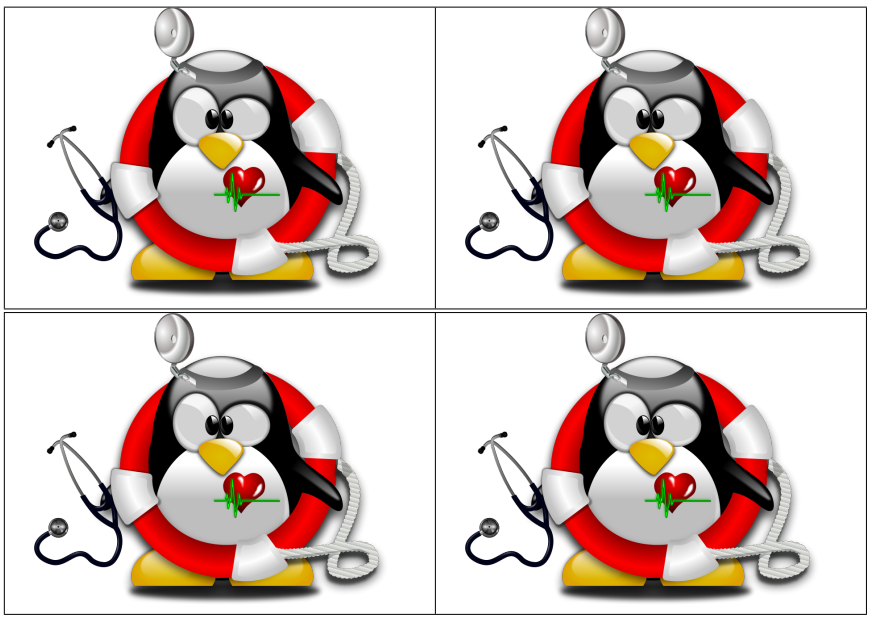
- 'labels' : to select the labels of the elements to display data,
- 'freq' : quiver frequencie, (default : 1)
- 'scale' : quiver scale, (default : ...)
- 'colordata' : set colors on each quiver (default : empty).

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`.

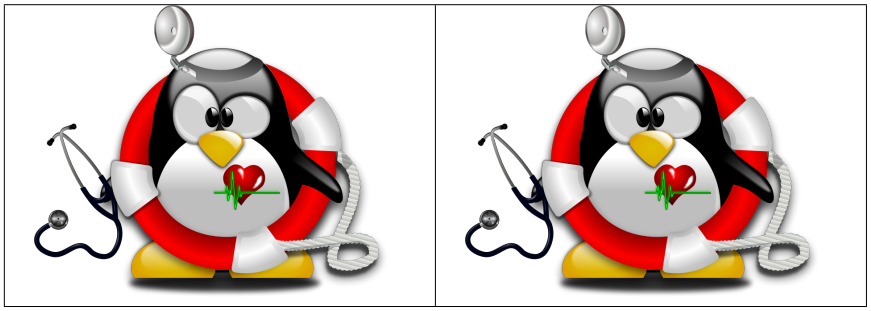


```
geofile=fc_simesh.get_geo(2,2,'condenser11');
meshfile=fc_oogmsh.buildmesh2d(geofile,25,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=@(x,y) cos(pi*x.^2).*cos(pi*y.^2);
U=Th.eval(u);
w={@(x,y) y.*cos(-(x.^2+y.^2)/10),@(x,y) -x.*cos(-(x.^2+y.^2)/10)};
W=Th.eval(w);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k',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

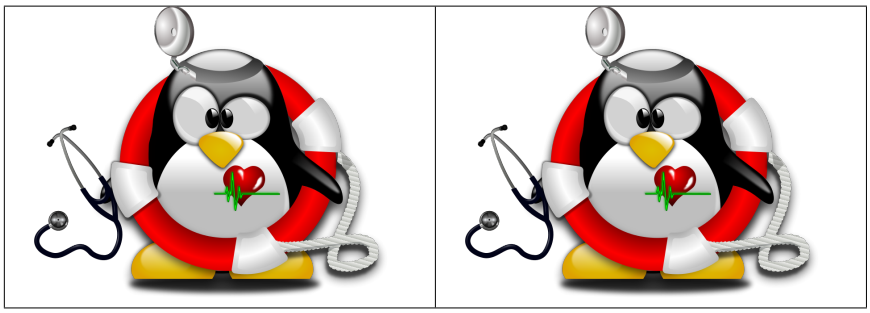


```
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5);
Th=fc_simesh.siMesh(meshfile);
w={@(x,y,z) y.*cos(-(x.^2+y.^2)/10),@(x,y,z) -x.*cos(-(x.^2+y.^2)/10),@(x,y,z) ...
    z/5 };
W=Th.eval(w);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k',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

10 fc_siplt.scatter function

The **fc_siplt**.scatter function displays scalar datas as colorized points on the mesh or parts of the mesh defined by an **fc_simesh.siMesh** object.

Syntaxe

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

Description

fc_siplt.scatter(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.

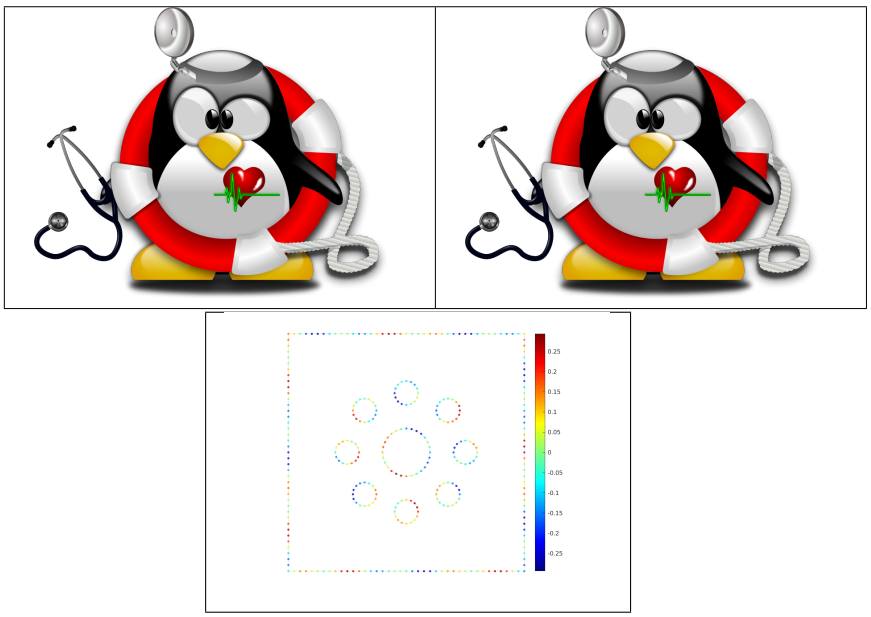
fc_siplt.scatter(Th,u,Name,Value, ...) specifies function options using one or more Name,Value pair arguments. Options of first level are

- 'd' : to specify the dimension of the simplices elements (default : Th.d)
- 'labels' : to select the labels of the elements to display data,
- 'MarkerSize' : size of the marker. Default is 1.

- 'ForcePatch' : if **true**, uses **patch** function, otherwise uses scatter function in dimension 2 or scatter3 function in dimension 3. Default is **false**.

The options of second level are those of the function used (see 'ForcePatch' option).

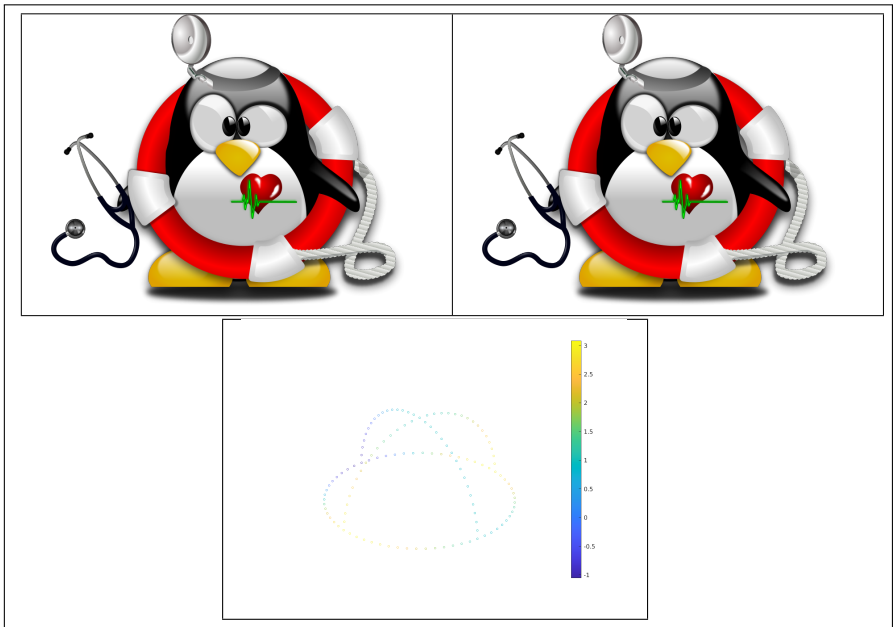
2D example : the following code is part of the **fc_siplt.demos.scatter2D** function.



```
meshfile=fc_oogmsh.gmsh.buildmesh2d('condenser11',20,varargin{:});
Th=fc_simesh.siMesh(meshfile);
u=@(x,y) 0.3*cos(5*(2*x-y)).*sin(5*(x-2*y));
U=Th.eval(u);

fc_tools.graphics.monitors.onGrid(2,2,'figures',1:3);

figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','LightGray')
hold on
fc_siplt.scatter(Th,U);
axis image;axis off;
colorbar

figure(2)
fc_siplt.plotmesh(Th,'d',1,'color','LightGray')
hold on
fc_siplt.scatter(Th,U,'MarkerSize',8);
axis off;axis image
colorbar

figure(3)
fc_siplt.plotmesh(Th,'d',1,'color','LightGray')
hold on  % '*Alpha' options not yet implemented under Octave
fc_siplt.scatter(Th,U,'d',1,'MarkerSize',12,'MarkerEdgeAlpha',0, ...
    'MarkerFaceColor','flat','MarkerFaceAlpha',0.8)
axis off;axis image
colormap('jet')
colorbar
```

Listing 18: Using fc_siplt.scatter function with a 2D mesh

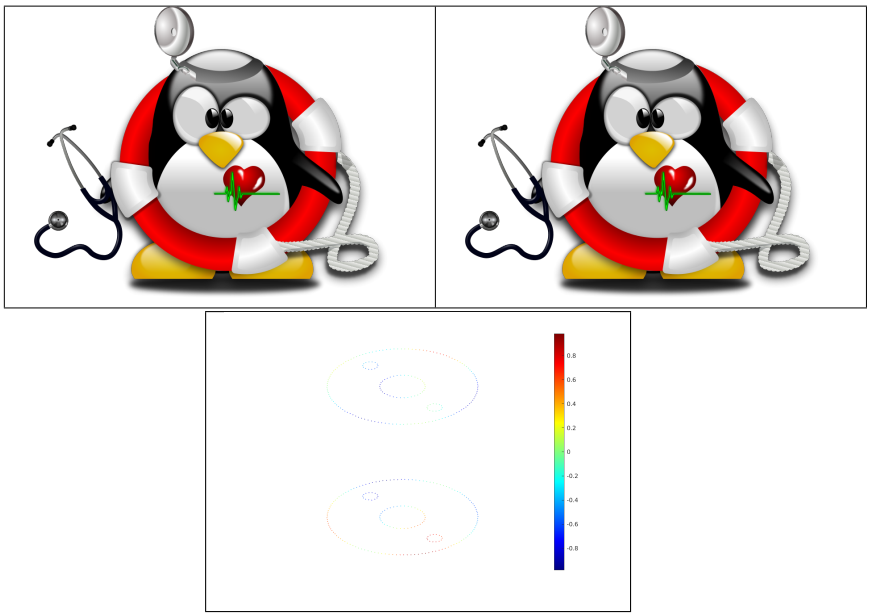
3Ds example : the following code is part of the **fc_siplt.demos.scatter3Ds** function.



```
geofile=fc_simesh.get_geo(3,2,'demisphere5');
meshfile=fc_oogmsh.buildmesh3ds(geofile,10,varargin{:});
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);

fc_tools.graphics.monitors.onGrid(2,2,'figures',1:3);

figure(1)
fc_siplt.scatter(Th,u,'MarkerSize',7)
view(3);axis off;axis image;colorbar;

figure(2)
fc_siplt.scatter(Th,u,'labels',[1,11])
hold on;axis off;axis image;colorbar;
fc_siplt.scatter(Th,u,'labels',[10,12], 'MarkerSize',9)
view(3)

figure(3)
fc_siplt.scatter(Th,u,'d',1,'MarkerSize',9)
axis off;axis image;colorbar
view(3)
```

Listing 19: Using fc_siplt.scatter function with a 3Ds mesh

3D example : the following code is part of the **fc_siplt.demos.scatter3D** function.



```
meshfile=fc_oogmsh.gmsh.buildmesh3d('cylinder3holes',20,varargin{:});
Th=fc_simesh.siMesh(meshfile);
u=@(x,y,z) cos(2*x-y-z).*sin(x-2*y+z);
U=Th.eval(u);

fc_tools.graphics.monitors.onGrid(2,2,'figures',1:3);

figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k')
hold on
fc_siplt.scatter(Th,U,'MarkerEdgeAlpha',0.1);
axis image;axis off;view(3)
colorbar

figure(2)
fc_siplt.plotmesh(Th,'d',1,'color','k')
hold on
fc_siplt.scatter(Th,U,'d',2);
axis image;axis off;view(3)
colorbar

figure(3)
fc_siplt.scatter(Th,U,'d',1);
axis image;axis off;view(3)
colormap('jet')
colorbar
```

Listing 20: Using fc_siplt.scatter function with a 3D mesh

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 : 0.2.5
commit : b4d6a9b05dccd6ac643d7235a057f89704c7e72a
  date : 2022-12-21
  time : 13-26-51
status : 0
------------------------------------------------
  name : fc-tools
   tag : 0.0.34.a
commit : 4795f7eaa711c3e2d1873157dd75c0ab2ed7d5b6
  date : 2023-01-05
  time : 10-23-14
status : 0
------------------------------------------------
  name : fc-bench
   tag : 0.1.3
commit : 6c8969bb49b90c7cae34e70c88dd2f968766376b
  date : 2022-12-17
  time : 13-58-43
status : 0
------------------------------------------------
  name : fc-hypermesh
   tag : 1.0.4
commit : 7f56fea428e57d2f30ae9d7b1f99bd6078807d23
  date : 2022-12-19
  time : 09-35-48
status : 0
------------------------------------------------
  name : fc-amat
   tag : 0.1.3
commit : 90dbaa2839188cf8e01629817e5f8ba1b7188052
  date : 2022-12-19
  time : 07-41-33
status : 0
------------------------------------------------
  name : fc-meshtools
   tag : 0.1.4
commit : 292552baea1b8dd1f3d58e73c68aca17999f68e2
  date : 2022-12-21
  time : 11-38-20
status : 0
------------------------------------------------
  name : fc-graphics4mesh
   tag : 0.1.5.a
commit : b994b7b2a57399848541708cae3c33f79787cb1a
  date : 2022-12-19
  time : 08-28-40
status : 0
------------------------------------------------
  name : fc-oogmsh
   tag : 0.2.4
commit : 8738562fc0b1cbdb028a8b58c5ba4f019e452831
  date : 2022-12-21
  time : 14-42-29
status : 0
------------------------------------------------
  name : fc-simesh
   tag : 0.4.5.a
commit : 86bb8660fa6384d6b68318f092fe778e7146b8bd
  date : 2023-01-05
  time : 10-22-05
status : 0
------------------------------------------------
```

git informations on the LaTeX package used to build this manual

```
--------------------------------------------------
   name : fctools
    tag :
 commit : c9a33ce7b4dacf90f66e5e49856e69afa1dac0a3
   date : 2022-12-17
   time : 07:57:20
 status : 1
--------------------------------------------------
```

Using the remote configuration repository:

url	`ssh://lagagit/MCS/Cuvelier/Matlab/fc-config`
commit	`268d29786e42ee37b3531aed5a97635dfc1d9ada`