



# fc siplt Octave package, User's Guide\* version 0.2.5

François Cuvelier<sup>†</sup>

January 5, 2023

## Abstract

This Octave package uses a `fc_simesh.siMesh` object, coming from the `fc_simesh` package, to display simplicial meshes or datas on simplicial meshes. Its kernel uses the `fc_graphics4mesh` package.

## 0 Contents

|          |                                   |          |
|----------|-----------------------------------|----------|
| <b>1</b> | <b>Introduction</b>               | <b>2</b> |
| <b>2</b> | <b>Installation</b>               | <b>4</b> |
| <b>3</b> | <b>fc_siplt.plotmesh function</b> | <b>5</b> |
| 3.1      | 2D example . . . . .              | 6        |
| 3.2      | 3D example . . . . .              | 7        |
| 3.3      | 3D surface example . . . . .      | 9        |

\* $\text{\LaTeX}$  manual, revision 0.2.5.a, compiled with Octave 7.3.0, and packages `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]

<sup>†</sup>Université Sorbonne Paris Nord, LAGA, CNRS, UMR 7539, F-93430, Villetaneuse, France, [cuvelier@math.univ-paris13.fr](mailto:cuvelier@math.univ-paris13.fr).

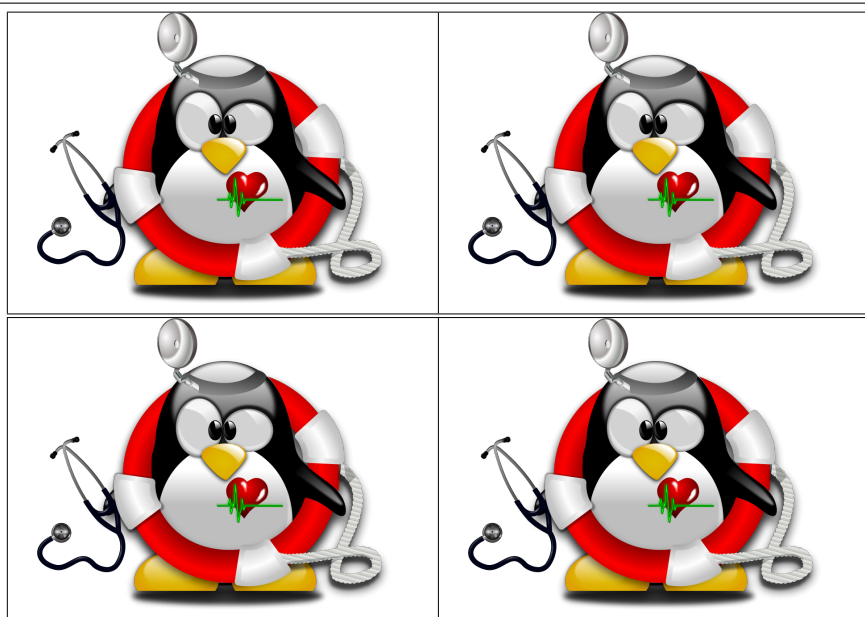
This work was supported by the ANR project DEDALES under grant ANR-14-CE23-0005.

|           |                                                  |           |
|-----------|--------------------------------------------------|-----------|
| <b>4</b>  | <b><code>fc_siplt.plot</code> function</b>       | <b>10</b> |
| 4.1       | 2D example . . . . .                             | 11        |
| 4.2       | 3D example . . . . .                             | 12        |
| 4.3       | 3D surface example . . . . .                     | 13        |
| <b>5</b>  | <b><code>fc_siplt.plotiso</code> function</b>    | <b>14</b> |
| 5.1       | 2D example . . . . .                             | 15        |
| 5.2       | 3D example . . . . .                             | 16        |
| 5.3       | 3D surface example . . . . .                     | 17        |
| <b>6</b>  | <b><code>fc_siplt.slicemesh</code> function</b>  | <b>18</b> |
| 6.1       | 3D example . . . . .                             | 19        |
| <b>7</b>  | <b><code>fc_siplt.slice</code> function</b>      | <b>19</b> |
| 7.1       | 3D example . . . . .                             | 20        |
| <b>8</b>  | <b><code>fc_siplt.sliceiso</code> function</b>   | <b>20</b> |
| 8.1       | 3D example . . . . .                             | 21        |
| <b>9</b>  | <b><code>fc_siplt.plotquiver</code> function</b> | <b>22</b> |
| 9.1       | 2D example . . . . .                             | 23        |
| 9.2       | 3D example . . . . .                             | 24        |
| 9.3       | 3D surface example . . . . .                     | 24        |
| <b>10</b> | <b><code>fc_siplt.scatter</code> function</b>    | <b>25</b> |
|           | <b>Appendices</b>                                | <b>29</b> |

## 1 Introduction

This **experimental** Octave package uses the [graphics4mesh](#) package[1] to do some graphic representations on a `fc_simesh.siMesh` object of the [simesh](#) package[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 package was only tested on Ubuntu 22.04.1 with Octave 7.3.0.

One just has 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* package and all the required packages 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 <fc-siplt> Octave package.
Copyright (C) 2017-2022 F. Cuvelier

1- Downloading and extracting the packages
2- Setting the <fc-siplt> package
Write in ~/Octave/fc-siplt-full/fc_siplt-0.2.5/configure_loc.m ...
3- Using packages :
->          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-oogmesh : 0.2.4
->          fc-simesh : 0.4.5
with          fc-siplt : 0.2.5
*** Using instructions
To use the <fc-siplt> package:
addpath('~/Octave/fc-siplt-full/fc_siplt-0.2.5')
fc_siplt.init()

See ~/Octave/ofc_siplt_set.m
```

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_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 ~/Octave/fc-siplt-full/fc_tools-0.0.34/configure_loc.m ...
Try to use default parameters!
  Use fc_bench.configure to configure.
Write in ~/Octave/fc-siplt-full/fc_bench-0.1.3/configure_loc.m ...
Try to use default parameters!
  Use fc_hypermesh.configure to configure.
Write in ~/Octave/fc-siplt-full/fc_hypermesh-1.0.4/configure_loc.m ...
Try to use default parameters!
  Use fc_amat.configure to configure.
Write in ~/Octave/fc-siplt-full/fc_amat-0.1.3/configure_loc.m ...
Try to use default parameters!
  Use fc_meshtools.configure to configure.
Write in ~/Octave/fc-siplt-full/fc_meshtools-0.1.4/configure_loc.m ...
Try to use default parameters!
  Use fc_graphics4mesh.configure to configure.
Write in ~/Octave/fc-siplt-full/fc_graphics4mesh-0.1.5/configure_loc.m ...
...
Try to use default parameters!
  Use fc_oogmsh.configure to configure.
Write in ~/Octave/fc-siplt-full/fc_oogmsh-0.2.4/configure_loc.m ...
Try to use default parameters!
  Use fc_simesh.configure to configure.
Write in ~/Octave/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

~/Octave/packages/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_sipplt.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).

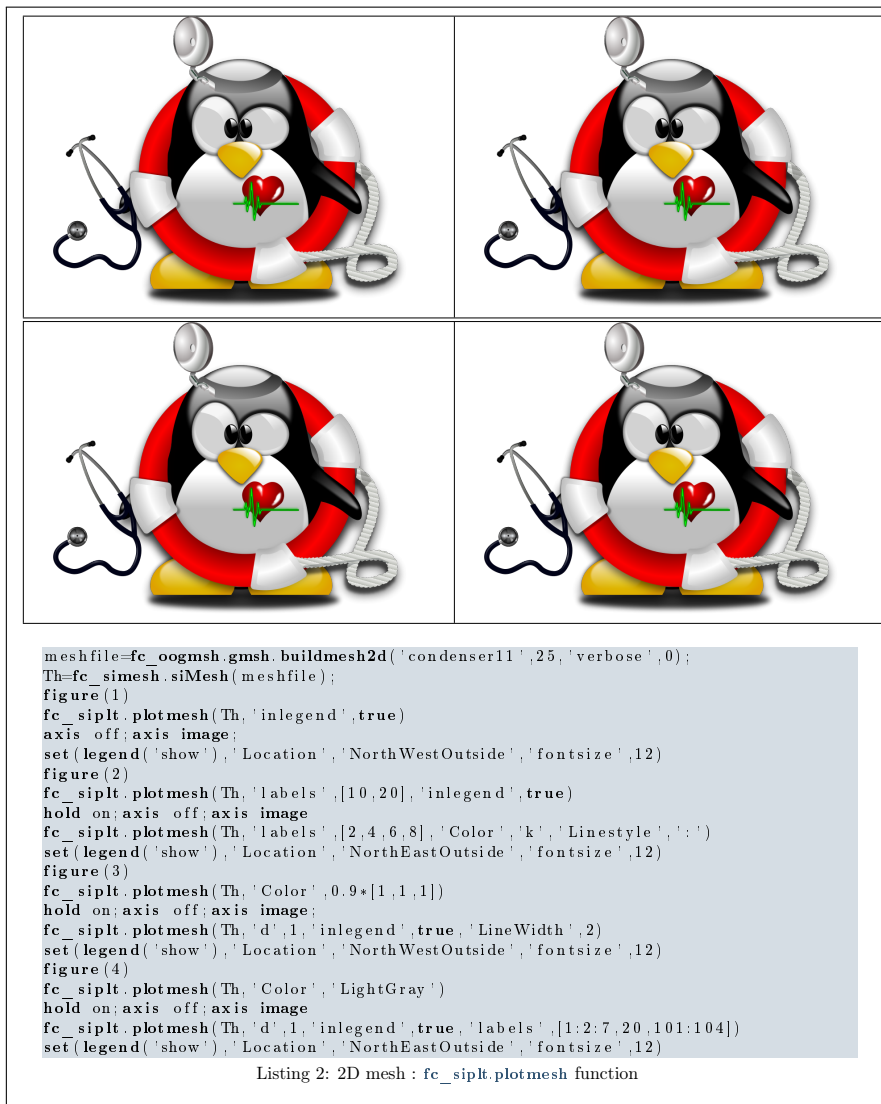
The options of second level depend on the type of elementaries mesh elements to represent.

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,

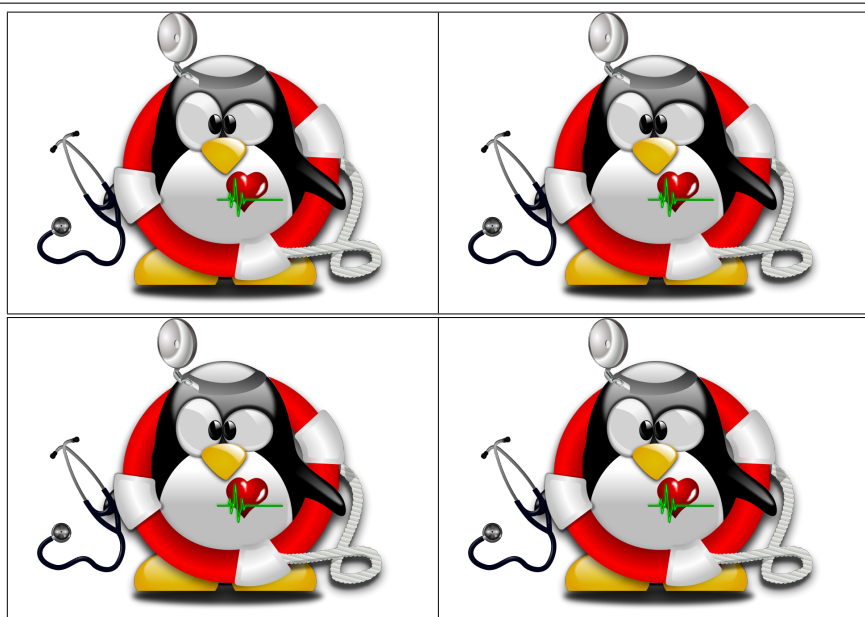
### 3.1 2D example

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



### 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.



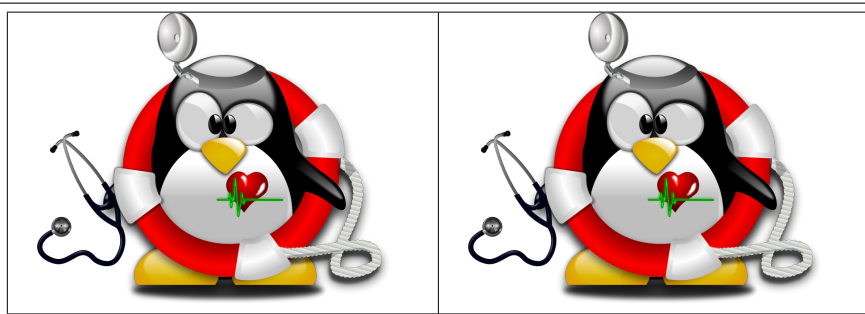
```

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





```

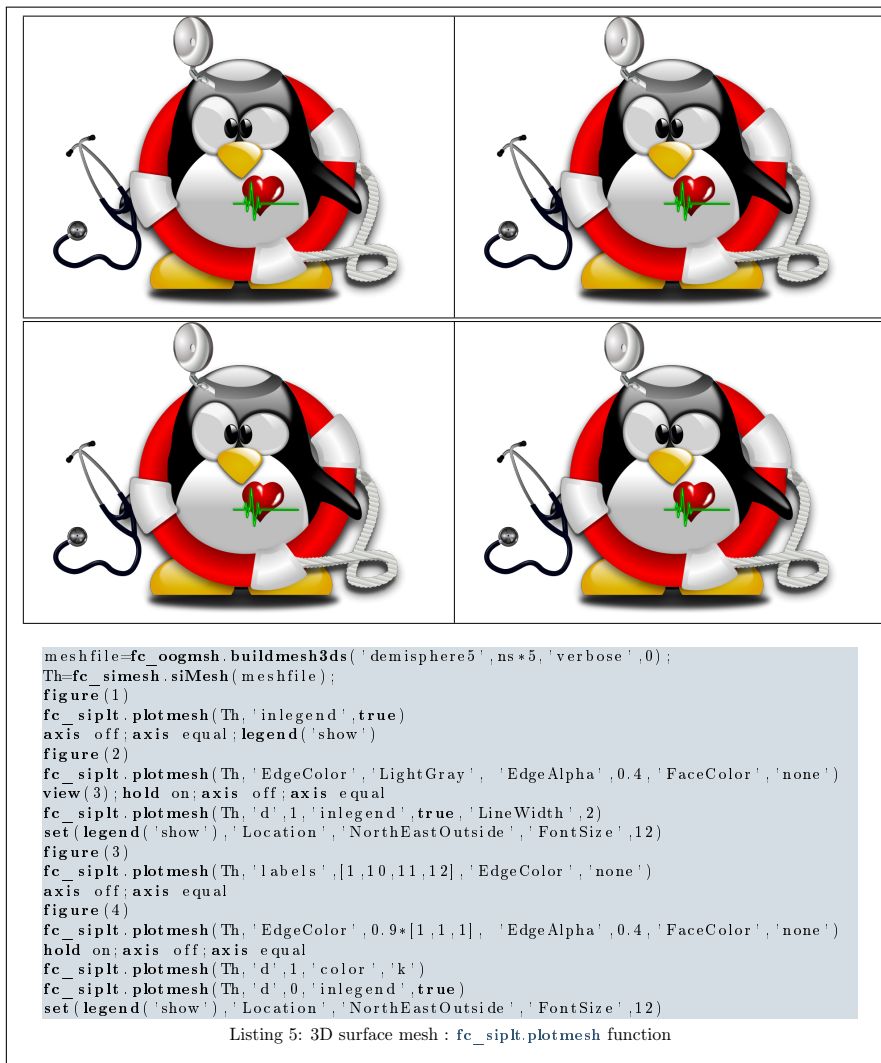
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.buildmesh3d(geofile,ns*5,'verbose',0);
figure(5)
P=[fc_tools.graphics.PlaneCoefs([0 0 1],[0 0 1]); ...
   fc_tools.graphics.PlaneCoefs([0 0 1],[-1 0 0])];
fc_siplt.plotmesh(Th,'d',1,'color','black')
hold on; axis image; view(-75,36)
fc_siplt.plotmesh(Th,'cutPlane',P,'Color','DarkGrey')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



## 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_sipplt.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`)

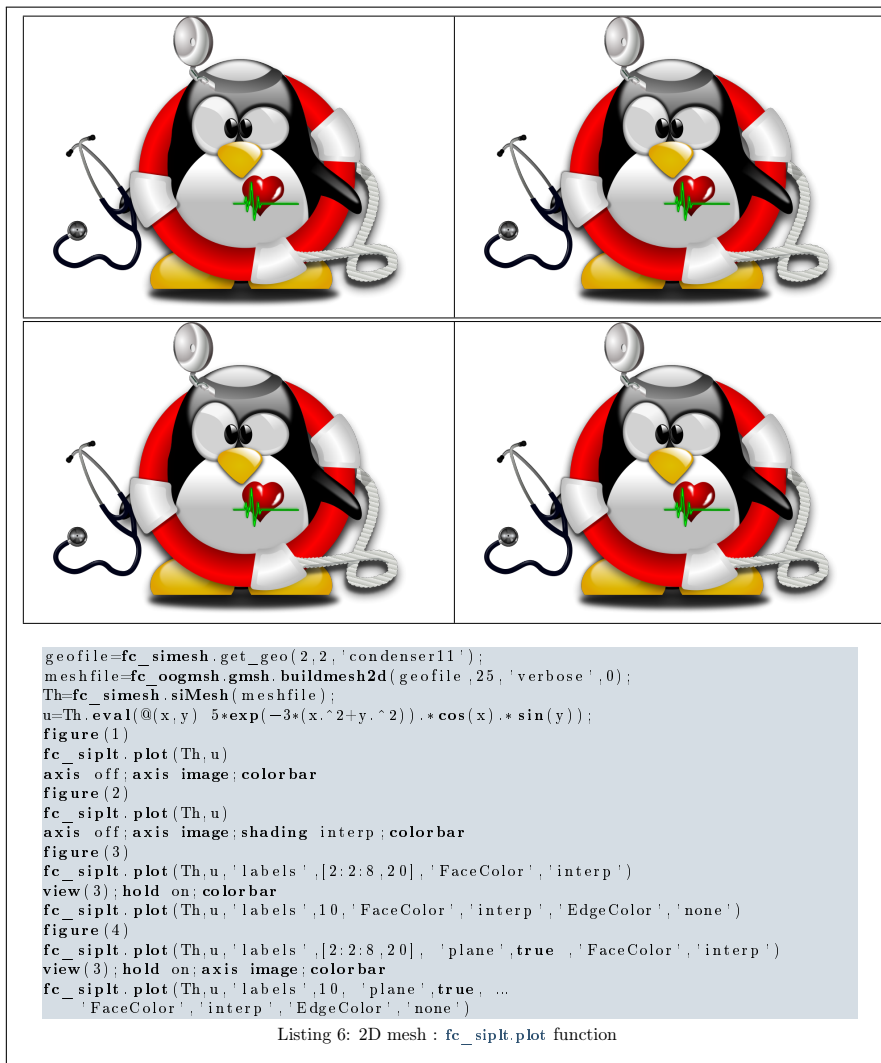
The options of second level depend on the type of elementaries mesh elements on which we want to represent datas.

One can use any option of the following functions according to the type of  $d$ -simplex.

- In dimension 3, `patch` function is used for  $d \in \llbracket 1, 3 \rrbracket$ .
- 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

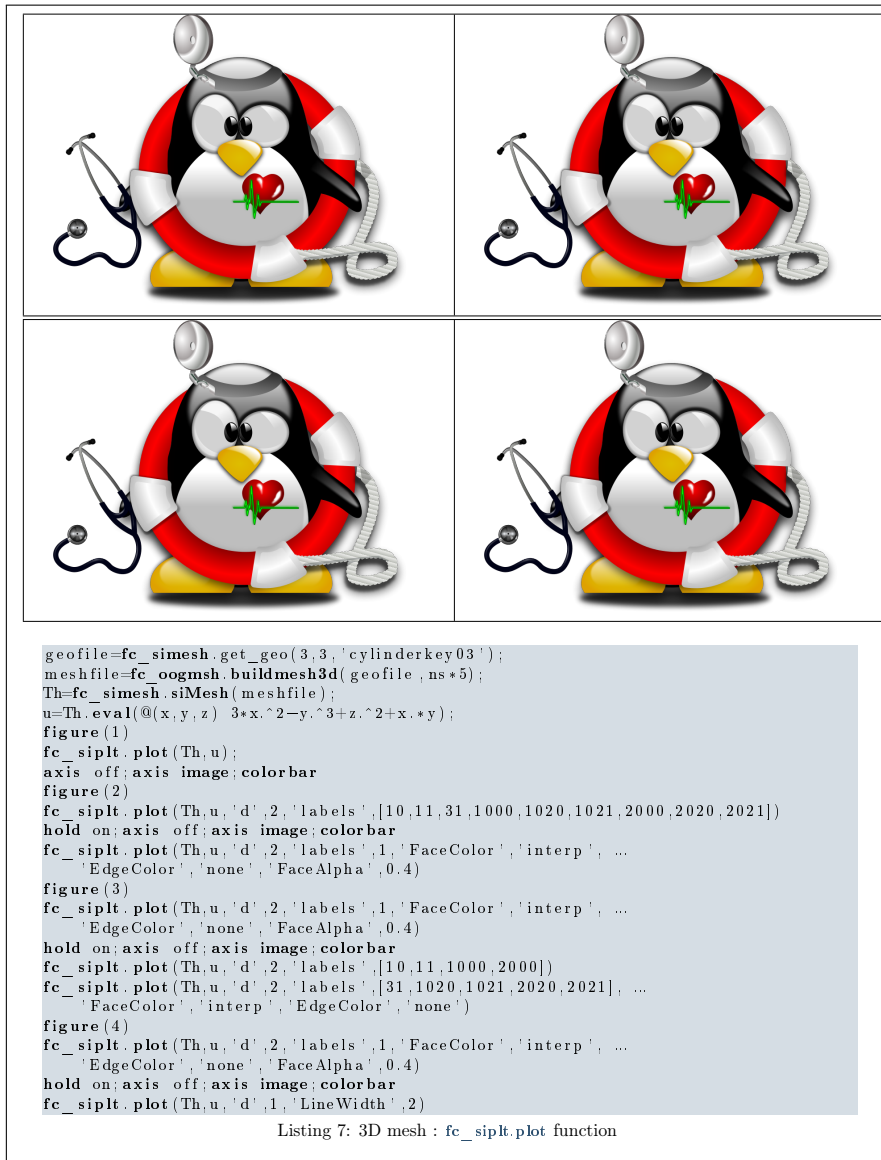
## 4.1 2D example

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



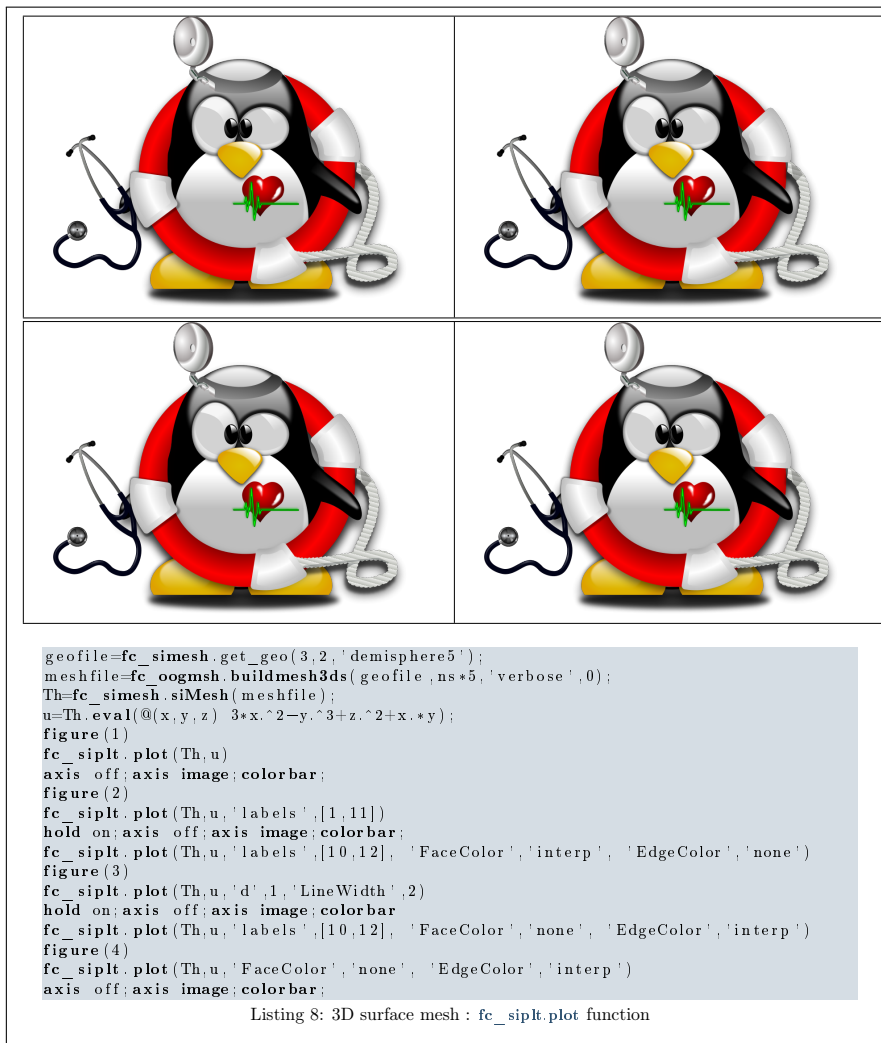
## 4.2 3D example

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.



### 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.



## 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

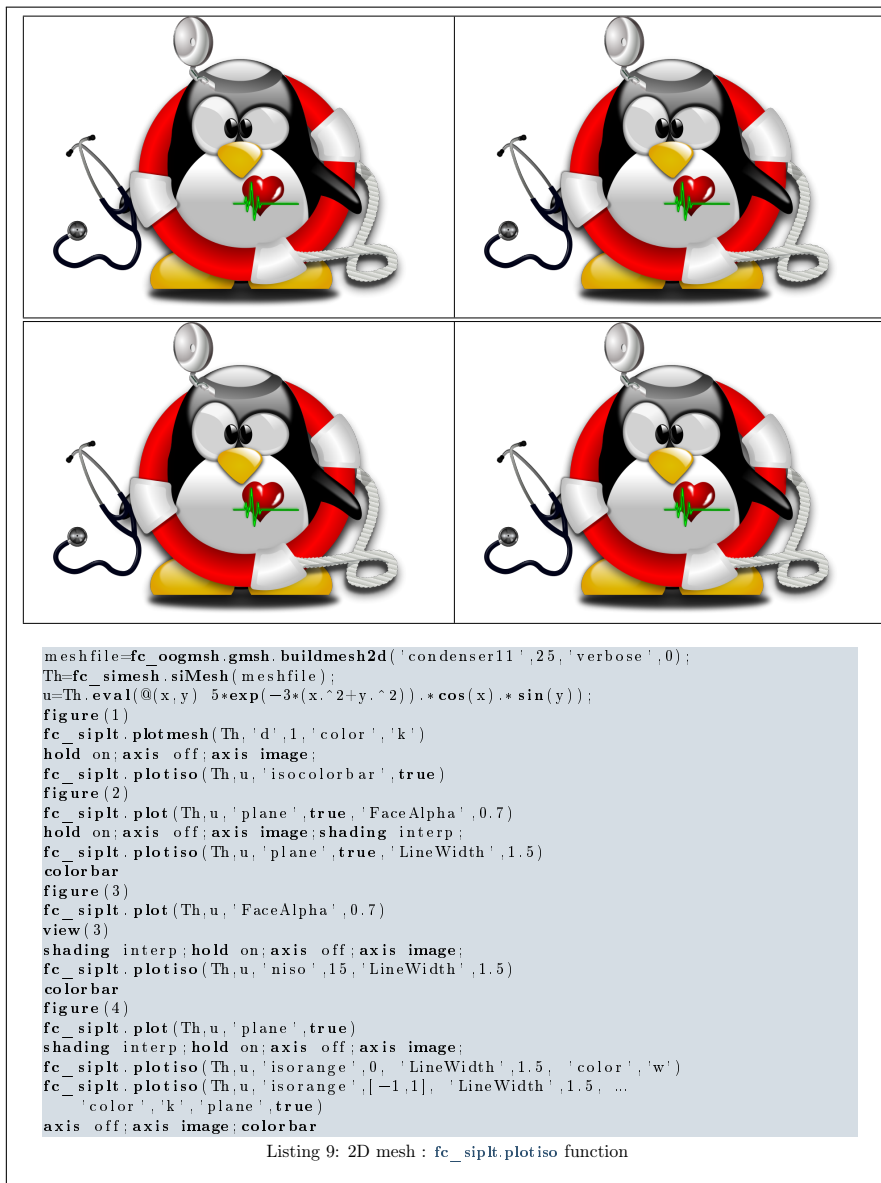
- **plot3** function in dimension 3 or in dimension 2 with `'plane'` option set to **false**
- **plot** function in 2 with `'plane'` option set to **true**

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

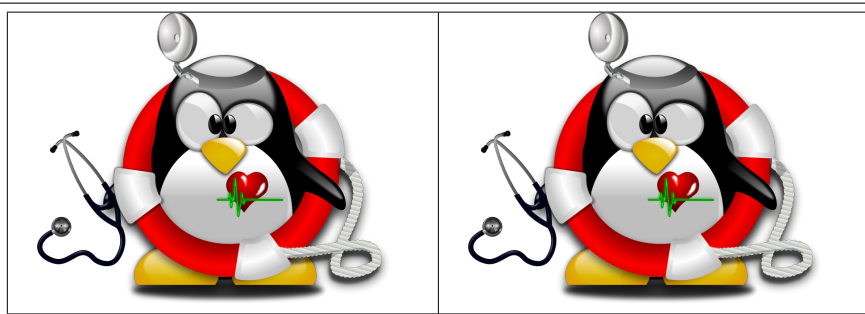
The following example use the *.geo* file `condenser11.geo` which is in the directory `geodir` of the toolbox.



## 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.





```

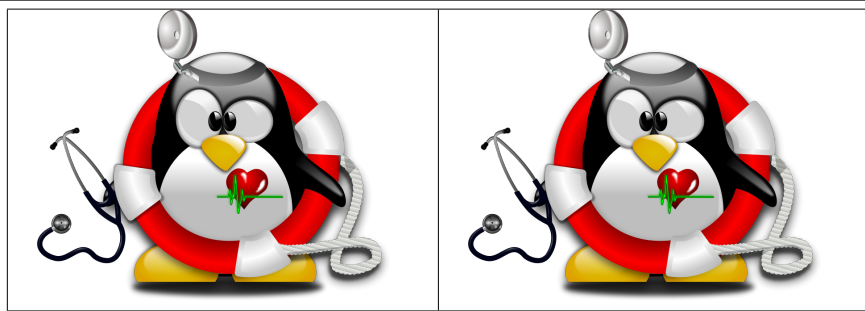
geofile=fc_simesh.get_geo(3,3,'cylinderkey03');
meshfile=fc_oogmsh.gmsh.buildmesh3d(geofile,ns*5,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
fc_siplt.plot(Th,u,'d',2,'labels',[10,11,31],'FaceColor','interp',...
'EdgeColor','none','FaceAlpha',0.4)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

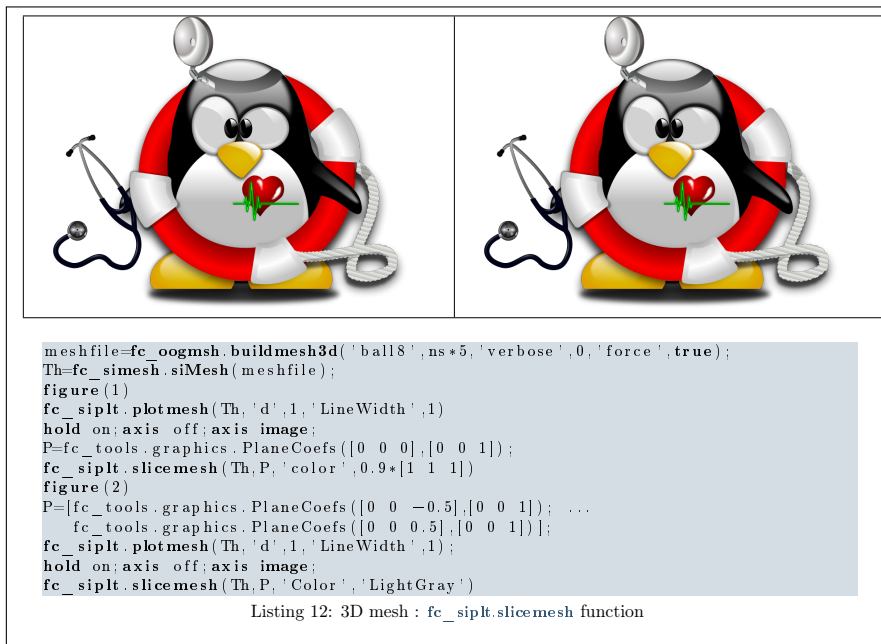
`fc_siplt.slicemesh(Th,P)` displays intersection of the plane defined by  $P(1)x + P(2)y + P(3)z + P(4) = 0$  and all the 3-dimensional simplices elements of `Th`, a `fc_simesh.siMesh` object. To compute  $P$  one can use the function `fc_tools.graphics.PlaneCoefs` of the [Ctools](#) package. With this 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 **ball18.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.



## 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

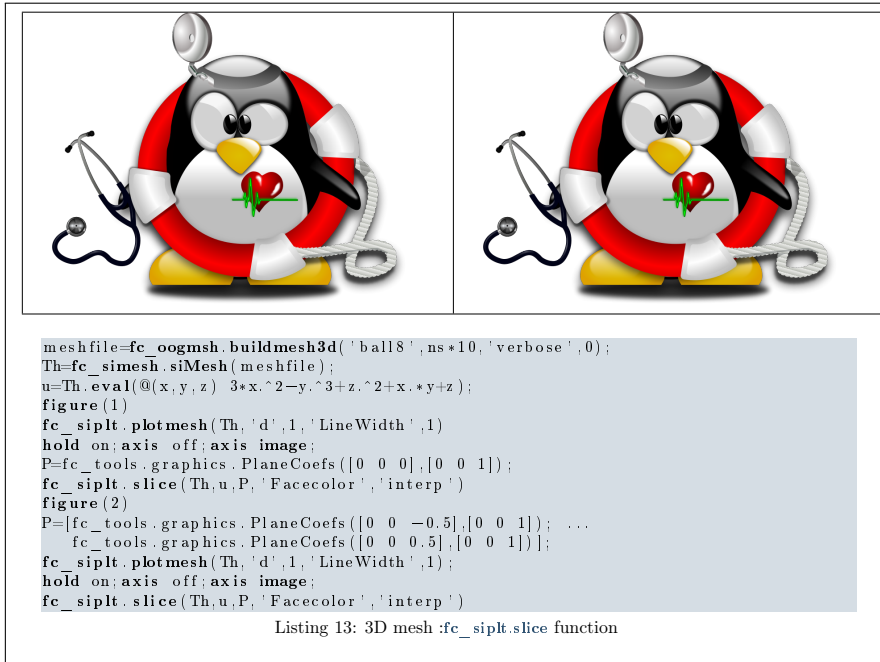
`fc_siplt.slice(Th,u,P)` displays `u` data on the intersection of the plane defined by  $P(1)x + P(2)y + P(3)z + P(4) = 0$  and all the 3-dimensional simplices elements of `Th`, a `fc_simesh.siMesh` object. The data `u` is an 1D-array of size `Th.nq` or `Th.nqGlobal` or `Th.nqParent`. To compute  $P$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



## 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

 package. With this function, the array  $P$ , is obtained with  $P = \text{fc\_tools.graphics.PlaneCoefs}(Q, V)$  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'.

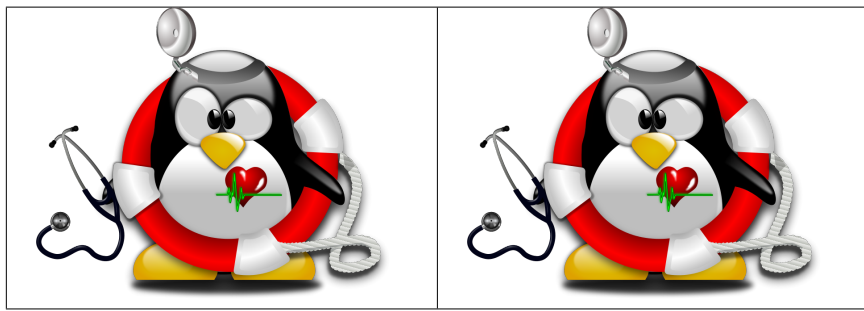
For key strings, one could also used any options of the plot3 function.

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 `ball18.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('ball18',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

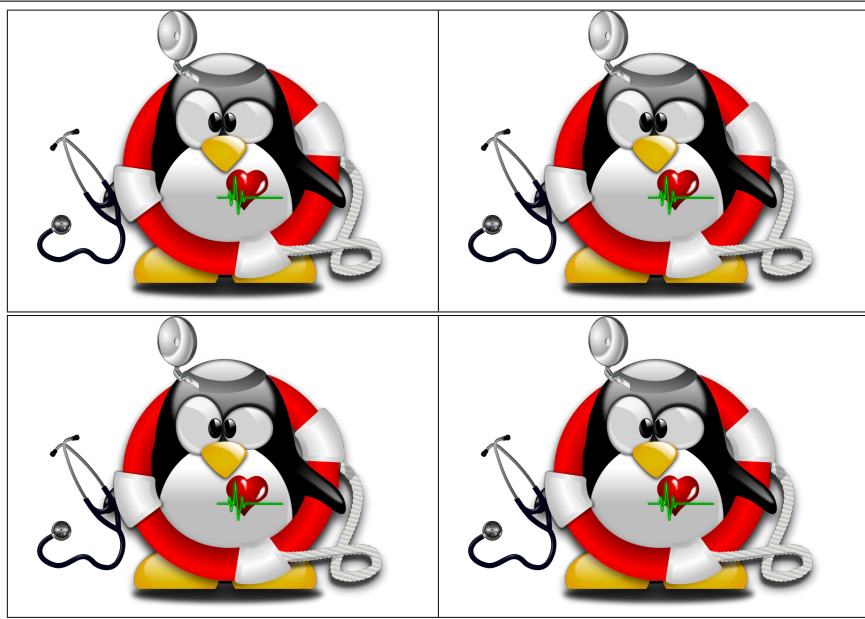
- 'labels' : to select the labels of the elements to display data,
- 'freq' : quiver frequencie, (default : 1)
- 'scale' : quiver scale, (default : ...)
- 'colordata' : set colors on each quiver (default : empty ).

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`.

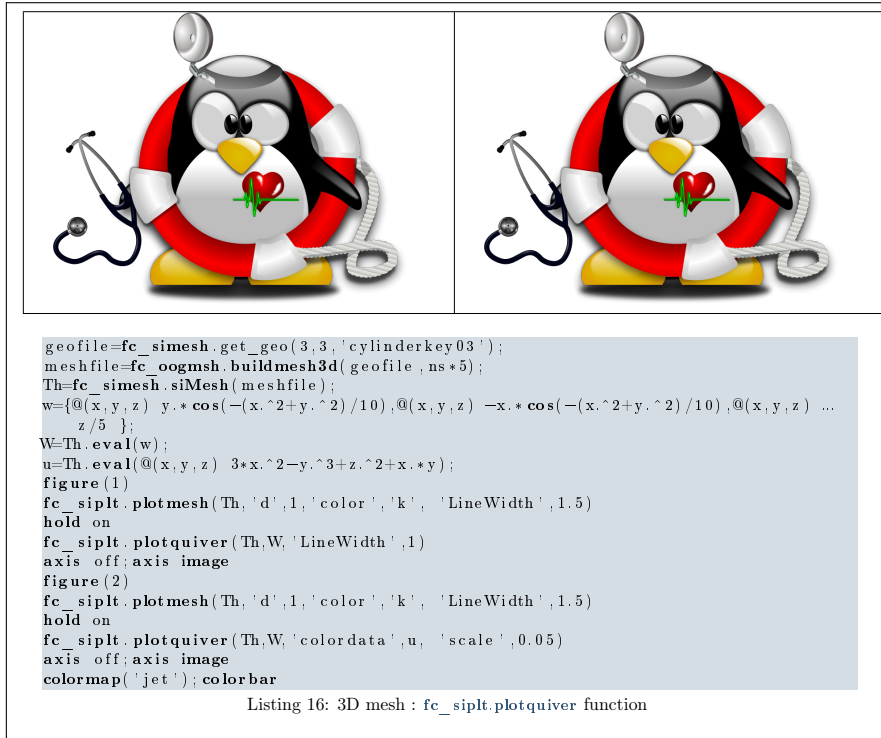


```
geofile=fc_simesh.get_geo(2,2,'condenser11');
meshfile=fc_oogmsh.buildmesh2d(geofile,25,'verbose',0);
Th=fc_simesh.siMesh(meshfile);
u=@(x,y) cos(pi*x.^2).*cos(pi*y.^2);
U=Th.eval(u);
w=@(x,y) y.*cos(-(x.^2+y.^2)/10),@(x,y) -x.*cos(-(x.^2+y.^2)/10);
W=Th.eval(w);
figure(1)
fc_siplt.plotmesh(Th,'d',1,'color','k','LineWidth',1.5)
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

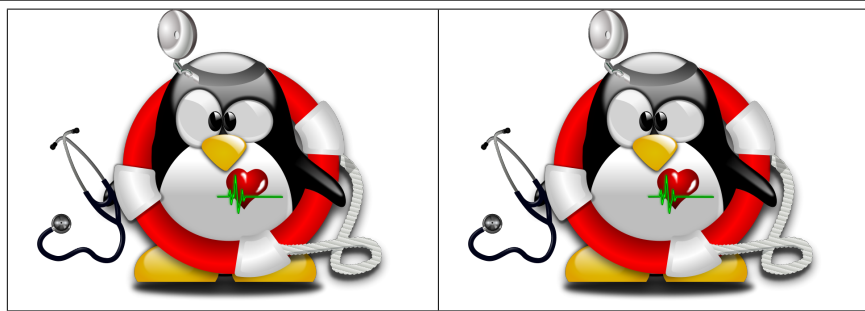
The following example use the *.geo* file `cylinderkey03.geo`. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



## 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`.

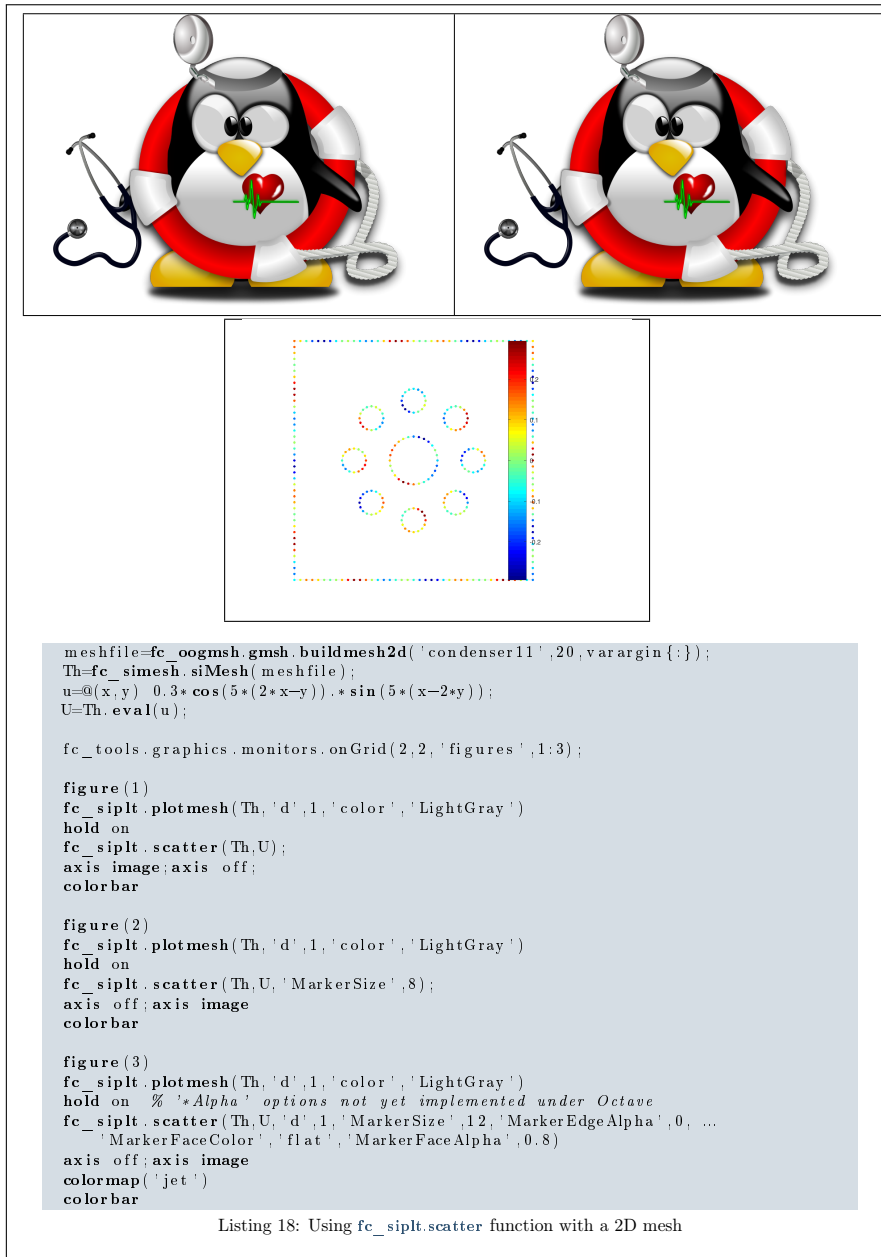
`fc_siplt.scatter(Th,u,Name,Value,...)` specifies function options using one or more `Name,Value` pair arguments. Options of first level are

- `'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.

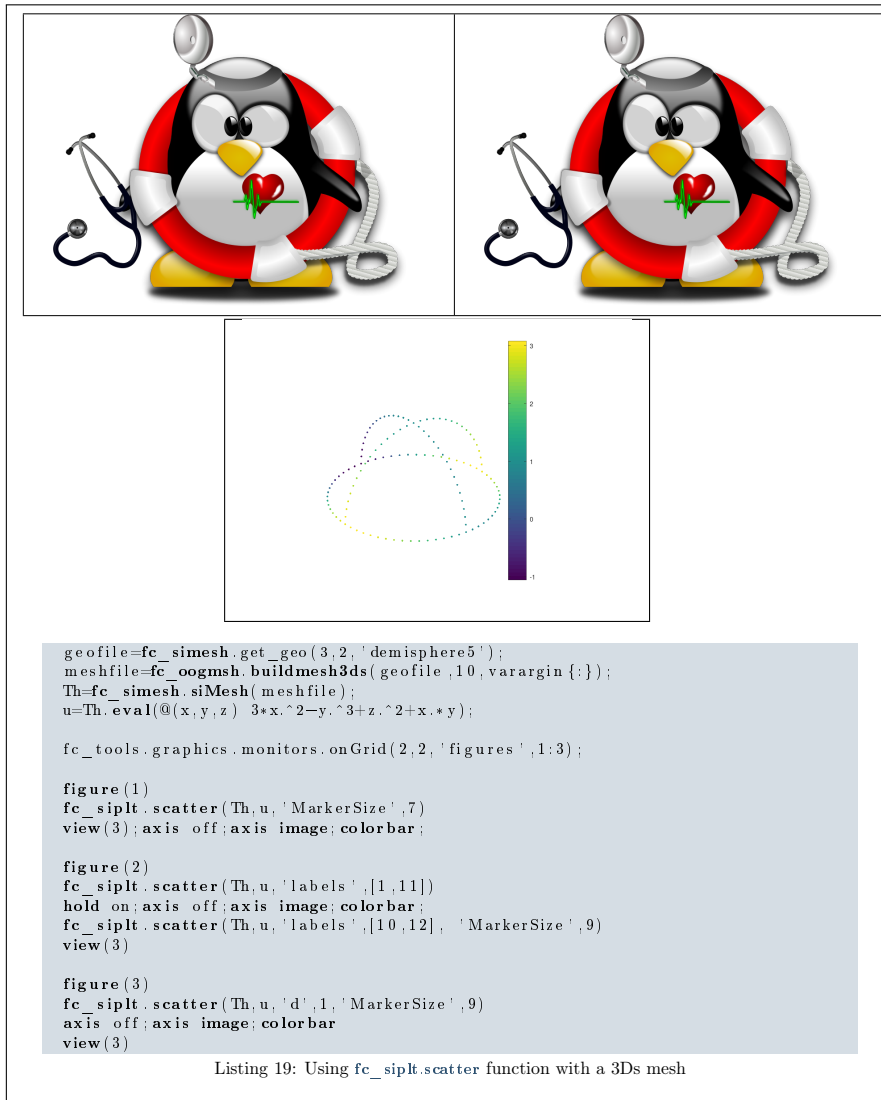
- 'ForcePatch' : if **true**, uses **patch** function, otherwise uses **scatter** function in dimension 2 or **scatter3** function in dimension 3. Default is **true**.

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

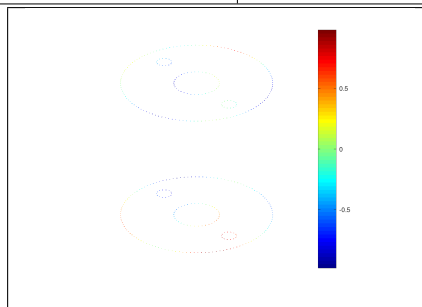
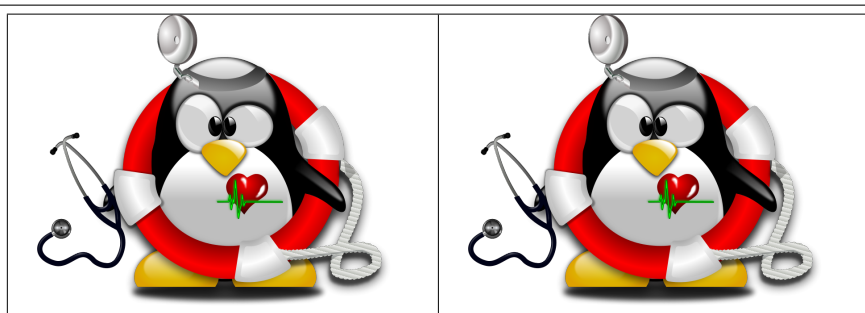
**2D example :** the following code is part of the **fc\_siplt.demos.scatter2D** function.



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



**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

|    |                                                                                                                                                   |    |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1  | <code>fc_siplt.demos.sample2D01</code> script with figure 1 (top left), figure 2 (top right), figure 3 (bottom left) and figure 4 (bottom right). | 3  |
| 2  | 2D mesh : <code>fc_siplt.plotmesh</code> function . . . . .                                                                                       | 7  |
| 3  | 3D plot mesh . . . . .                                                                                                                            | 8  |
| 4  | 3D mesh : <code>fc_siplt.plotmesh</code> function . . . . .                                                                                       | 9  |
| 5  | 3D surface mesh : <code>fc_siplt.plotmesh</code> function . . . . .                                                                               | 10 |
| 6  | 2D mesh : <code>fc_siplt.plot</code> function . . . . .                                                                                           | 12 |
| 7  | 3D mesh : <code>fc_siplt.plot</code> function . . . . .                                                                                           | 13 |
| 8  | 3D surface mesh : <code>fc_siplt.plot</code> function . . . . .                                                                                   | 14 |
| 9  | 2D mesh : <code>fc_siplt.plotiso</code> function . . . . .                                                                                        | 16 |
| 10 | 3D mesh : <code>fc_siplt.plotiso</code> function . . . . .                                                                                        | 17 |
| 11 | 3D surface mesh : <code>fc_siplt.plotiso</code> function . . . . .                                                                                | 18 |
| 12 | 3D mesh : <code>fc_siplt.slicemesh</code> function . . . . .                                                                                      | 19 |
| 13 | 3D mesh : <code>fc_siplt.slice</code> function . . . . .                                                                                          | 20 |
| 14 | 3D mesh : <code>fc_siplt.sliceiso</code> function . . . . .                                                                                       | 22 |
| 15 | 2D mesh : <code>fc_siplt.plotquiver</code> function . . . . .                                                                                     | 23 |
| 16 | 3D mesh : <code>fc_siplt.plotquiver</code> function . . . . .                                                                                     | 24 |
| 17 | 3D surface mesh : <code>fc_siplt.plotquiver</code> function . . . . .                                                                             | 25 |
| 18 | Using <code>fc_siplt.scatter</code> function with a 2D mesh . . . . .                                                                             | 26 |
| 19 | Using <code>fc_siplt.scatter</code> function with a 3Ds mesh . . . . .                                                                            | 27 |
| 20 | Using <code>fc_siplt.scatter</code> function with a 3D mesh . . . . .                                                                             | 28 |

## B References

- [1] F. Cuvelier. `fc_graphics4mesh`: an Octave package 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 Octave package 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



# Octave package

git informations on the packages used to build this manual

```
-----
name : fc-siplt
tag : 0.2.5
commit : b4d6a9b05dcd6ac643d7235a057f89704c7e72a
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 : 292552bae1b8dd1f3d58e73c68aca17999f68e2
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
-----
[fc-tools] waiting 2(s) to finish saving figures
```

git informations on the L<sup>A</sup>T<sub>E</sub>X package used to build this manual

```
-----  
name : fctools  
tag :  
commit : c9a33ce7b4daf90f66e5e49856e69afaldac0a3  
date : 2022-12-17  
time : 07:57:20  
status : 1  
-----  
[fc-tools] waiting 2(s) to finish saving figures
```

Using the remote configuration repository:

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