FC-PSIMESH Octave package, User's Guide *

François Cuvelier†

February 8, 2017

Abstract

This object-oriented Octave package allows to use partitioned simplices meshes generated from `gmsh` (in dimension 2 or 3). A particular care was taken to the graphics representations of partitioned meshes and datas on partitioned meshes.

This code uses the FC-SIMESHOctave package and we suppose the reader to be familiar with it.

Contents

1 Partitioned mesh, PSIMESH object **2**
1.1 Samples . . . 3
1.1.1 2D samples . . . 3
1.1.2 3D samples . . . 4
1.1.3 3D surface samples . . . 9
1.2 PSIMESH object . . . 10

2 PSIMESH methods **11**
2.1 PSIMESH constructor . . . 11
2.2 function EVAL . . . 11
2.2.1 2D example . . . 12
2.2.2 3D example . . . 13
2.2.3 3D surface example . . . 13
2.3 function PLOTMESH . . . 14

*Compiled with Octave 4.2.0

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

This work was partially supported by ANR Dedales.

2.3.1 2D example . . . 15
2.3.2 3D example . . . 19
2.3.3 3D surface example . . . 22
2.4 function PLOT . . . 24
2.4.1 2D example . . . 25
2.4.2 3D example . . . 27
2.4.3 3D surface example . . . 30
2.4.4 function PLOTISO . . . 31
2.4.5 2D example . . . 31
2.4.6 3D example . . . 32
2.4.7 3D surface example . . . 34
2.4.8 function SLICEMESH . . . 34
2.4.9 3D example . . . 35
2.4.10 function SLICE . . . 36
2.4.11 3D example . . . 37
2.4.12 function SLICEISO . . . 37
2.4.13 3D example . . . 38
2.4.14 function PLOTVECTORFIELD . . . 38
2.4.15 2D example . . . 39
2.4.16 3D example . . . 41
2.4.17 3D surface example . . . 43

1 Partitioned mesh, psiMesh object

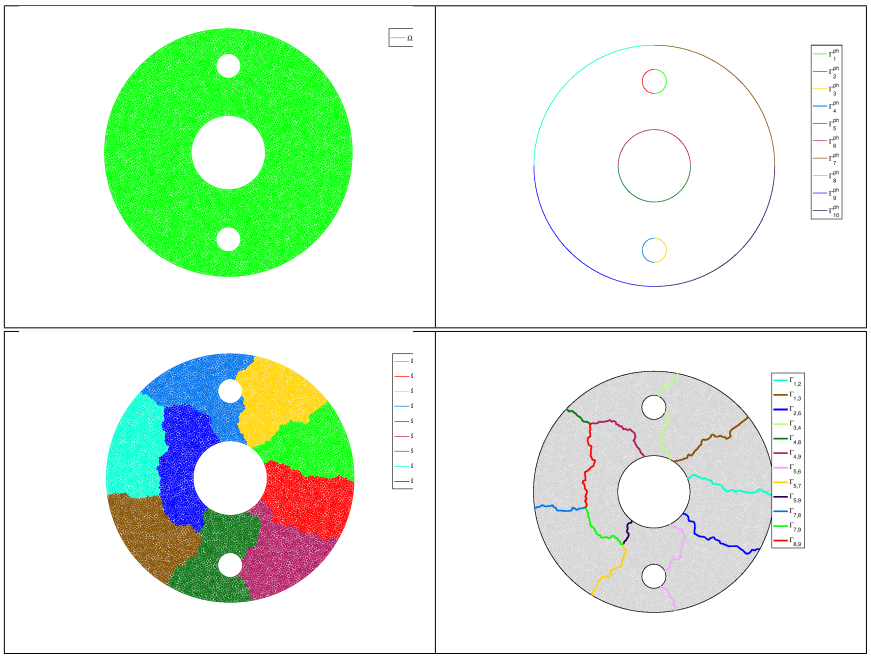
In geometry, a simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions. Specifically, a k-simplex in $\mathbb{R}^{\dim}$, $k \leqslant \dim$, is a polytope which is the convex hull of its $k+1$ vertices of $\mathbb{R}^{\dim}$. More formally, suppose the $k+1$ vertices $\mathrm{q}^0, \cdots, q^k \in \mathbb{R}^{\dim}$ such that $\mathrm{q}^1 - \mathrm{q}^0, \cdots, \mathrm{q}^k - \mathrm{q}^0$ are linearly independent. Then, the k-simplex K determined by them is the set of points

$$K = \left\{ \sum_{i=0}^{k} \lambda_i \mathrm{q}^i \mid \lambda_i \geqslant 0,\ i \in [\![0,k]\!],\ \text{with} \sum_{i=0}^{k} \lambda_i = 1 \right\}.$$

- A **k-simplicial elementary mesh** in $\mathbb{R}^{\dim}$, $k \leqslant \dim$, is a mesh with **unique label** only composed with k-simplices. It corresponds to the SIMESHELT object of the FC-SIMESH package.
- A d-**simplicial mesh** in $\mathbb{R}^{\dim}$, $\mathrm{d} \leqslant \dim$, is an union of **$k$-simplicial elementary meshes** with $k \in [\![0,\mathrm{d}]\!]$. It corresponds to the SIMESH object of the FC-SIMESH package.
- A **partitioned** d-**simplicial mesh** in dimension dim, is an union of d-**simplicial mesh** in $\mathbb{R}^{\dim}$. It corresponds to PSIMESH object defined in this package.

1.1 Samples

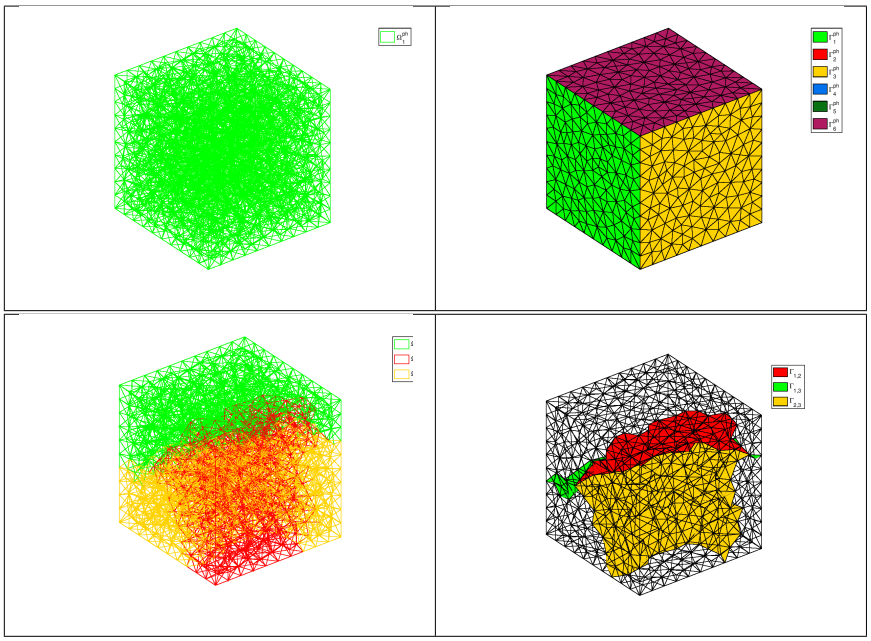
1.1.1 2D samples



```
meshfile=gmsh.buildmesh2d('disk3holes',50);
meshpartfile=gmsh.buildpartmesh2d(meshfile,9);
pTh=psiMesh(meshpartfile);
figure(1)
pTh.plotmesh('physicals','legend',true)
axis off;axis image
figure(2)
pTh.plotmesh('physicals','d',1,'legend',true)
axis off;axis image
figure(3)
pTh.plotmesh('legend',true)
axis off;axis image
figure(4)
pTh.plotmesh('color','LightGrey')
hold on;axis off;axis image
pTh.plotmesh('physicals','d',1,'color','k')
pTh.plotmesh('interfaces','legend',true,'linewidth',2)
```

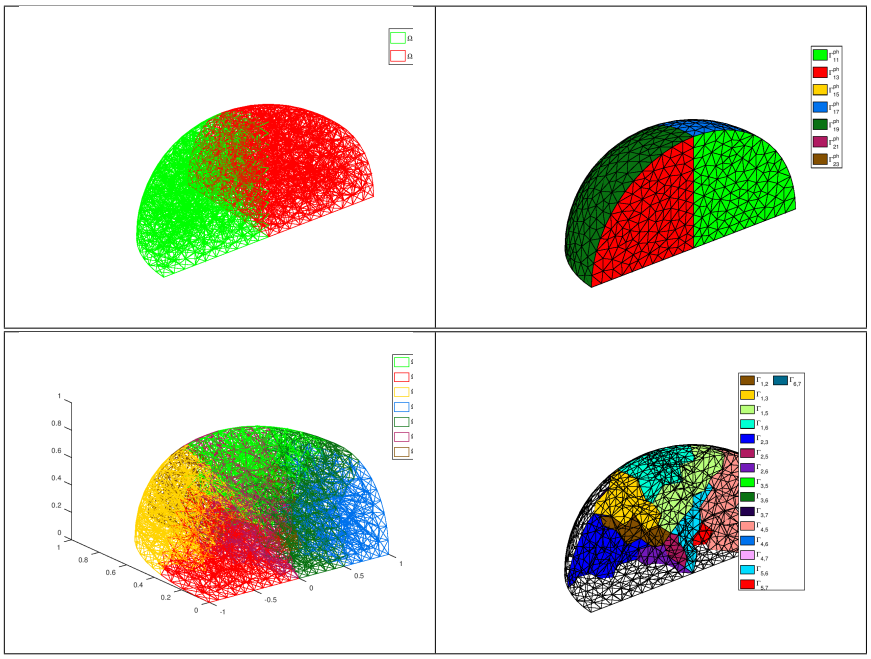
Listing 1: Mesh from `disk3holes.geo`, label of the domain (upper left), labels of the boundaries (upper right), labels of the partitions (bottom left) and labels of the interfaces (bottom right)

1.1.2 3D samples



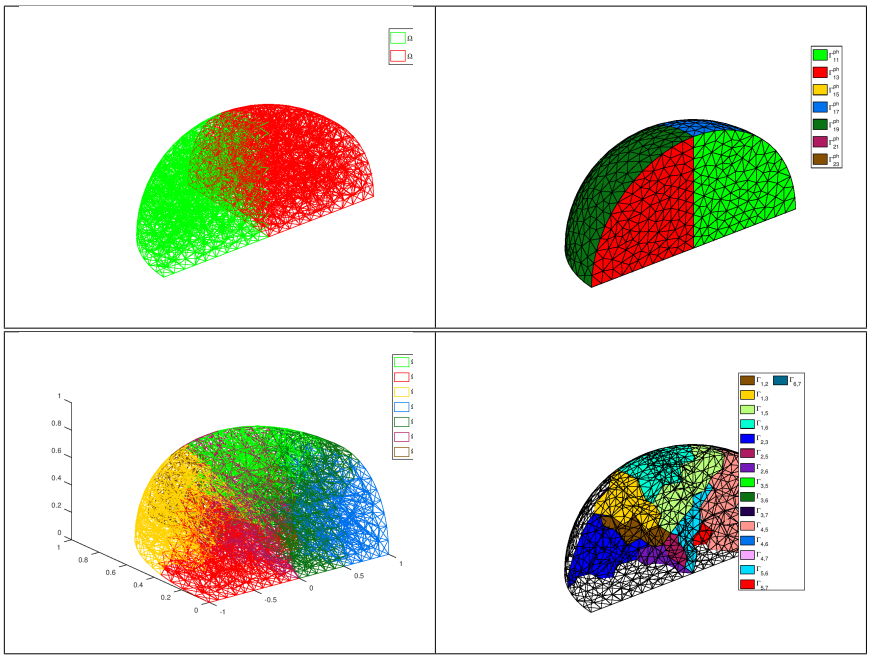
```
meshfile=gmsh.buildmesh3d('cube6',10);
meshpartfile=gmsh.buildpartmesh3d(meshfile,3);
pTh=psiMesh(meshpartfile);
figure(1)
pTh.plotmesh('physicals','legend',true)
axis off;axis image
figure(2)
pTh.plotmesh('physicals','d',2,'legend',true)
axis off;axis image
figure(3)
pTh.plotmesh('legend',true)
axis off;axis image
figure(4)
pTh.plotmesh('physicals','d',2,'faceColor',[0.8,0.8,0.8], ...
    'faceAlpha',0.2)
hold on
pTh.plotmesh('interfaces','legend',true)
axis off;axis image
```

Listing 2: Mesh from `cube6.geo`, label of the domain (upper left), labels of the boundaries (upper right), labels of the partitions (bottom left) and labels of the interfaces (bottom right)



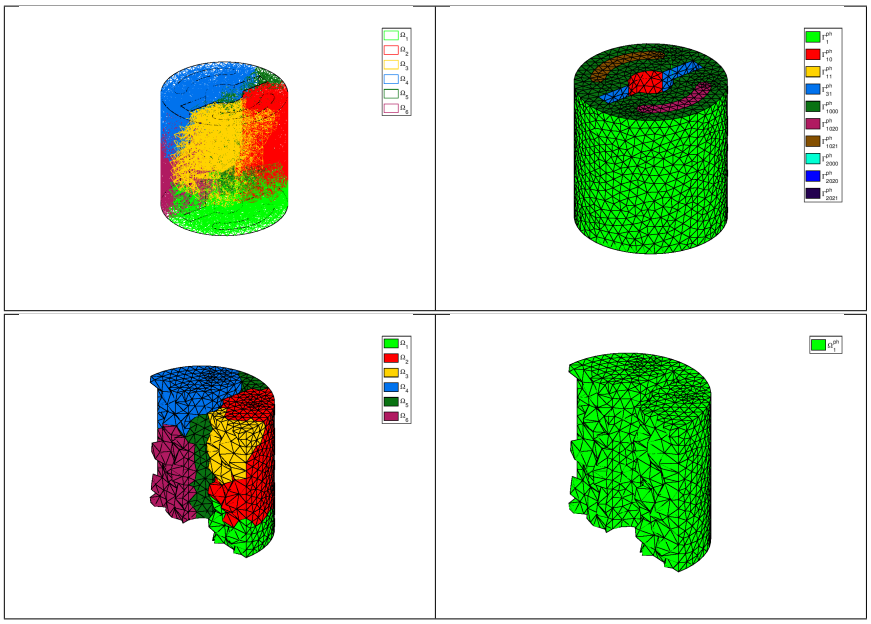
```
meshfile=gmsh.buildmesh3d('quart_sphere2',5);
meshpartfile=gmsh.buildpartmesh3d(meshfile,7);
pTh=psiMesh(meshpartfile);
figure(1)
pTh.plotmesh('physicals','legend',true)
figure(2)
pTh.plotmesh('physicals','d',2,'legend',true)
figure(3)
pTh.plotmesh('legend',true)
figure(4)
pTh.plotmesh('physicals','d',2, ...
    'faceColor',[0.8,0.8,0.8], 'faceAlpha',0.2)
hold on
pTh.plotmesh('interfaces','legend',true)
```

Listing 3: Mesh from `quart_sphere2.geo`, labels of the domains (upper left), labels of the boundaries (upper right), labels of the partitions (bottom left) and labels of the interfaces (bottom right)



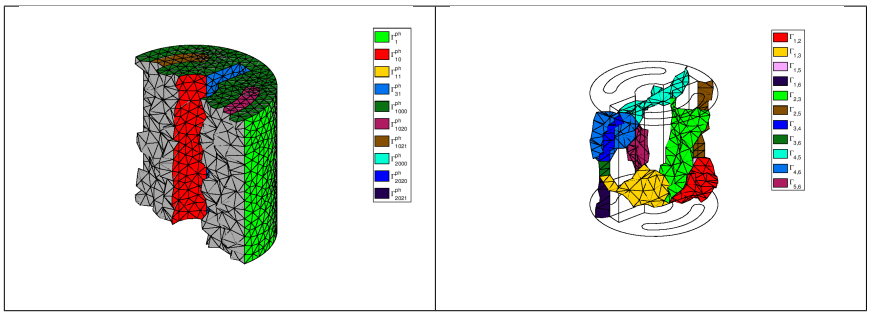
```
meshfile=gmsh.buildmesh3d('quart_sphere2',5);
meshpartfile=gmsh.buildpartmesh3d(meshfile,7);
pTh=psiMesh(meshpartfile);
figure(1)
pTh.plotmesh('physicals','legend',true)
figure(2)
pTh.plotmesh('physicals','d',2,'legend',true)
figure(3)
pTh.plotmesh('legend',true)
figure(4)
pTh.plotmesh('physicals','d',2, ...
    'faceColor',[0.8,0.8,0.8], 'faceAlpha',0.2)
hold on
pTh.plotmesh('interfaces','legend',true)
```

Listing 4: Mesh from `quart_sphere2.geo`, labels of the domains (upper left), labels of the boundaries (upper right), labels of the partitions (bottom left) and labels of the interfaces (bottom right)



```
pTh=fc_psimesh.build_psimesh('cylinderkey03','N',ns*8,'np',6, ...
    'geodir',geodir3d, 'd',3,'verbose',2);
figure(1)
pTh.plotmesh('partitions','labels',1,'legend',true)
axis off;axis equal;
hold on
pTh.plotmesh('d',1,'Color','black');
figure(2)
pTh.plotmesh('physicals','d',2,'legend',true)
axis equal;
P=fc_tools.graphics.PlaneCoefs([0 0 0],[-1 0 0]);
figure(3)
pTh.plotmesh('cutPlan',P,'legend',true)
axis off;axis equal;
figure(4)
pTh.plotmesh('physicals','cutPlan',P,'legend',true)
axis off;axis equal;
```

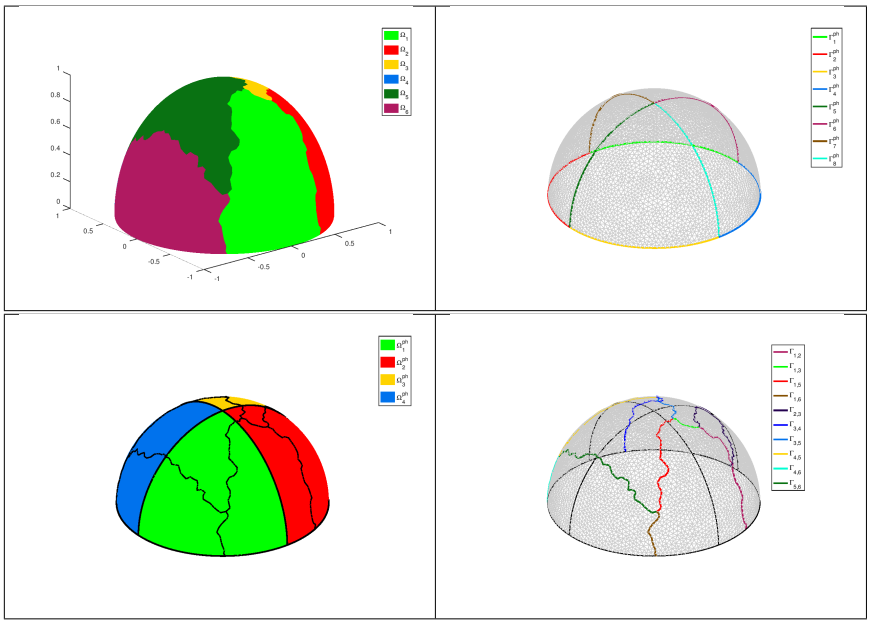
Listing 5: Mesh from `cylinderkey03.geo`, with 6 partitions



```
pTh=fc_psimesh.build_psimesh('cylinderkey03','N',ns*8,'np',6, ...
    'geodir',geodir3d, 'd',3,'verbose',2);
P=fc_tools.graphics.PlaneCoefs([0 0 0],[-1 0 0]);
figure(1)
pTh.plotmesh('cutPlan',P,'color','DarkGrey')
axis off;axis equal;
hold on
pTh.plotmesh('physicals','d',2,'cutPlan',P,'legend',true)
figure(2)
pTh.plotmesh('d',1,'color','black')
axis off;axis equal;
hold on
pTh.plotmesh('interfaces','d',2,'legend',true)
```

Listing 6: Mesh from `cylinderkey03.geo` with 6 partitions (more)

1.1.3 3D surface samples



```
meshfile=gmsh.buildmesh3ds('demisphere4surf',ns*25,'force',false);
meshpartfile=gmsh.buildpartmesh3ds(meshfile,6,'force',false);
pTh=psiMesh(meshpartfile);
figure(1);
pTh.plotmesh('EdgeColor','none','legend',true);
figure(2)
pTh.plotmesh('FaceColor','None' ...
    ,'EdgeColor',0.8*[1,1,1],'EdgeAlpha',0.3);
hold on
pTh.plotmesh('physicals','d',1,'LineWidth',2,'legend',true);
figure(3)
pTh.plotmesh('physicals','EdgeColor','None','legend',true);
hold on
pTh.plotmesh('physicals','d',1,'color',[0,0,0], ...
    'LineWidth',2);
pTh.plotmesh('interfaces','Color',[0,0,0], ...
    'LineWidth',1.5, 'LineStyle','-.');
figure(4)
pTh.plotmesh('FaceColor','None', ...
    'EdgeColor',0.8*[1,1,1], 'EdgeAlpha',0.3);
hold on
pTh.plotmesh('physicals','d',1,'color','k', 'LineWidth',1);
pTh.plotmesh('interfaces','LineWidth',2,'legend',true);
```

Listing 7: Mesh from `demisphere4surf.geo`, labels of the domains (upper left), labels of the boundaries (upper right), labels of the partitions (bottom left) and labels of the interfaces (bottom right)

1.2 PSIMESH object

Partitioned simplicial mesh, PSIMESH object

dim	:	integer space dimension
d	:	integer d-dimensional simplicial mesh
$\mathcal{T}_h$	:	array of SIMESH objects, one by partition we set $\Omega_i = \mathcal{T}_h(i)$
np	:	number of partitions (i.e. length of $\mathcal{T}_h$)
n_q	:	number of vertices
`PhysElts`	:	cell arrays of physical/geometrical structure ...
`InterElts`	:	cell arrays of interfaces structure ...

`InterElts`, interface element structure associated with an PSIMESH object

d	:	integer d-dimensional simplicial interfaces
labels	:	list of interface labels in PSIMESH object
parts	:	np-by-1 cells array. $parts\{i\}$ contains all interface labels of $\Omega_i = \mathcal{T}_h(i)$
`Gamma`	:	np-by-np (sparse or not) matrix. `Gamma`(i,j) is the interface label of $\partial\Omega_i \cap \partial\Omega_j$ if not empty, 0 otherwise. We set $\Gamma_{i,j}$ =`Gamma`(i,j)
`GammaIdx`	:	np-by-np (sparse or not) matrix. `GammaIdx`(i,j) 0 if $\Gamma_{i,j} = \varnothing$ otherwise `GammaIdx`(i,j) is the index in the cells array $\mathcal{T}_h(i).sTh$ of $\Gamma_{i,j}$.

`PhysElts`, physical element structure associated with an PSIMESH object

d	:	integer d-dimensional simplicial physical elements
labels	:	list of nl physical labels in PSIMESH object
parts	:	array of np lists parts(i) contains all physical labels of $\Omega_i = \mathcal{T}_h(i)$
connect	:	nl-by-np sparse matrix. connect(l,i)$\neq$ 0 if labels(l) physical element is partially in Ω_i

2 psiMesh methods

2.1 psiMesh constructor

The constructor of the `psiMesh` class can initialize the object from a partitioned mesh file (obtained from `gmsh`).

> **remark 2.1**
>
> The interfaces beetween mesh partitions must have their labels less than 0 or, greater than or equal to 10000.

Syntaxe

```
pTh=psiMesh(meshfile)
```

Description

`pTh=psiMesh(meshfile)` create the `psiMesh` object `pTh` from the mesh file `meshfile` (`gmsh` format by default).

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

Octave commands with output

```
meshfile=gmsh.buildmesh2d('condenser11',25,'verbose',0);
mpfile=gmsh.buildpartmesh2d(meshfile,5,'verbose',0);
disp('*** Read partitioned mesh ***')
pTh=psiMesh(mpfile);
info(pTh)
```

```
*** Read partitioned mesh ***
Variable pTh [psiMesh object] :
  dim=2, d=2, np=5
  nq=2990, nme=5722
Partition 1 , label 1 :
  nq=623, nme=1143
Partition 2 , label 2 :
  nq=624, nme=1144
Partition 3 , label 3 :
  nq=631, nme=1145
Partition 4 , label 4 :
  nq=625, nme=1145
Partition 5 , label 5 :
  nq=633, nme=1145
```

2.2 function EVAL

The method `eval` evaluates a function on a partitioned mesh defined by an `psiMesh` object.

Syntaxe

```
pTh.eval(fun)
pTh.eval(fun,Name,Value, ...)
```

Description

pTh.**eval**(fun) evaluates the function fun on all the vertices of the partitioned mesh. The return value is an array of dimension Th.nq-by-1. If fun is a cell array of m functions, the return value is a pTh.nq-by-m array.

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

- `'split'` : the return value is splitted by partitions and so its a cell array of dimension pTh.np where each cell (one by partition) is an array of dimension the number of vertices of the partition by m.

2.2.1 2D example

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

Octave commands with output

```
pTh=fc_psimesh.build_psimesh('condenser11','N',25,'np',7);
u=@(x,y) cos(x+y);
pTh.info()
disp('without_splitting')
U=pTh.eval(u);
V=pTh.eval(u,'split',true);
whos('U','V')
```

```
[fc-oogmsh] Input file : <fc-oogmsh>/geodir/condenser11.geo
[fc-oogmsh] Mesh file <fc-oogmsh>/meshes/condenser11-25.msh already exists.
  -> Use "force" flag to rebuild if needed.
[fc-oogmsh] Input file : <fc-oogmsh>/meshes/condenser11-25.msh
[fc-oogmsh] Mesh file <fc-oogmsh>/meshes/condenser11-25-part7.msh already exists.
  -> Use "force" flag to rebuild if needed.
Variable [psiMesh object] :
  dim=2, d=2, np=7
  nq=2990, nme=5722
Partition 1 , label 1 :
  nq=449, nme=817
Partition 2 , label 2 :
  nq=454, nme=818
Partition 3 , label 3 :
  nq=454, nme=817
Partition 4 , label 4 :
  nq=455, nme=817
Partition 5 , label 5 :
  nq=450, nme=818
Partition 6 , label 6 :
  nq=450, nme=817
Partition 7 , label 7 :
  nq=456, nme=818
without splitting
Variables in the current scope:

   Attr Name      Size                Bytes  Class
   ==== ====      ====                =====  =====
        U       2990x1                23920  double
        V          7x1                25344  cell

Total is 2997 elements using 49264 bytes
```

2.2.2 3D example

The following example use the *.geo* file `cylinderkey03.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.

Octave commands with output

```
geofile='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
u=@(x,y,z) cos(x+y-z);
pTh.info()
U=pTh.eval(u);
V=pTh.eval(u,'split',true);
whos('U','V')
```

```
[fc-oogmsh] Input file : /home/cuvelier/Travail/Recherch/Matlab/fc-psimesh/doc/../geodir/3d/cylinderkey03.geo
[fc-oogmsh] Mesh file <fc-oogmsh>/meshes/cylinderkey03-8.msh already exists.
  -> Use "force" flag to rebuild if needed.
[fc-oogmsh] Input file : <fc-oogmsh>/meshes/cylinderkey03-8.msh
[fc-oogmsh] Mesh file <fc-oogmsh>/meshes/cylinderkey03-8-part6.msh already exists.
  -> Use "force" flag to rebuild if needed.
Variable [psiMesh object] :
  dim=3, d=3, np=6
  nq=3025, nme=12029
Partition 1 , label 1 :
  nq=598, nme=2004
Partition 2 , label 2 :
  nq=564, nme=2005
Partition 3 , label 3 :
  nq=560, nme=2005
Partition 4 , label 4 :
  nq=574, nme=2005
Partition 5 , label 5 :
  nq=582, nme=2005
Partition 6 , label 6 :
  nq=551, nme=2005
Variables in the current scope:

   Attr Name      Size                Bytes  Class
   ==== ====      ====                =====  =====
        U       3025x1                24200  double
        V          6x1                27432  cell

Total is 3031 elements using 51632 bytes
```

2.2.3 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 mesh with simplices of dimensions 1 and 2.

Octave commands with output

```
pTh=fc_psimesh.build_psimesh('demisphere4surf','N',ns*10, ...
    'np',6,'d',2);
u=@(q) cos(q(1,:)+q(2,:)-q(3,:));
pTh.info()
U=pTh.eval(u);
V=pTh.eval(u,'split',true);
whos('U','V')
```

```
[fc-oogmsh] Input file : <fc-oogmsh>/geodir/demisphere4surf.geo
[fc-oogmsh] Mesh file <fc-oogmsh>/meshes/demisphere4surf-10.msh already exists.
  -> Use "force" flag to rebuild if needed.
[fc-oogmsh] Input file : <fc-oogmsh>/meshes/demisphere4surf-10.msh
[fc-oogmsh] Mesh file <fc-oogmsh>/meshes/demisphere4surf-10-part6.msh already exists.
  -> Use "force" flag to rebuild if needed.
Variable [psiMesh object] :
  dim=3, d=2, np=6
  nq=794, nme=1522
Partition 1 , label 1 :
  nq=150, nme=253
Partition 2 , label 2 :
  nq=150, nme=254
Partition 3 , label 3 :
  nq=148, nme=254
Partition 4 , label 4 :
  nq=148, nme=253
Partition 5 , label 5 :
  nq=153, nme=255
Partition 6 , label 6 :
  nq=152, nme=253
Variables in the current scope:

   Attr Name      Size                Bytes  Class
   ==== ====      ====                =====  =====
        U        794x1                 6352  double
        V          6x1                 7208  cell

Total is 800 elements using 13560 bytes
```

2.3 function PLOTMESH

The method plotmesh displays completes or partials *kinds* of mesh elements defined by an `psiMesh` object. The select *kind* of mesh elements to draw is one of the following :

'partitions' (default), 'physicals' or 'interfaces'

Syntaxe

```
pTh.plotmesh()
pTh.plotmesh(kind)
pTh.plotmesh(Name,Value, ...)
pTh.plotmesh(kind,Name,Value, ...)
```

Description

`pTh.plotmesh()` displays all the pTh.d-dimensional simplices elements of all the partitions.

`pTh.plotmesh(kind)` where kind is the string 'partitions' (default), 'physicals' or 'interfaces'. Displays all the d-dimensional simplices elements of all the kind contains in `psiMesh` object pTh. The default value for d is pTh.d−1 if kind is 'interfaces' otherwise is pTh.d.

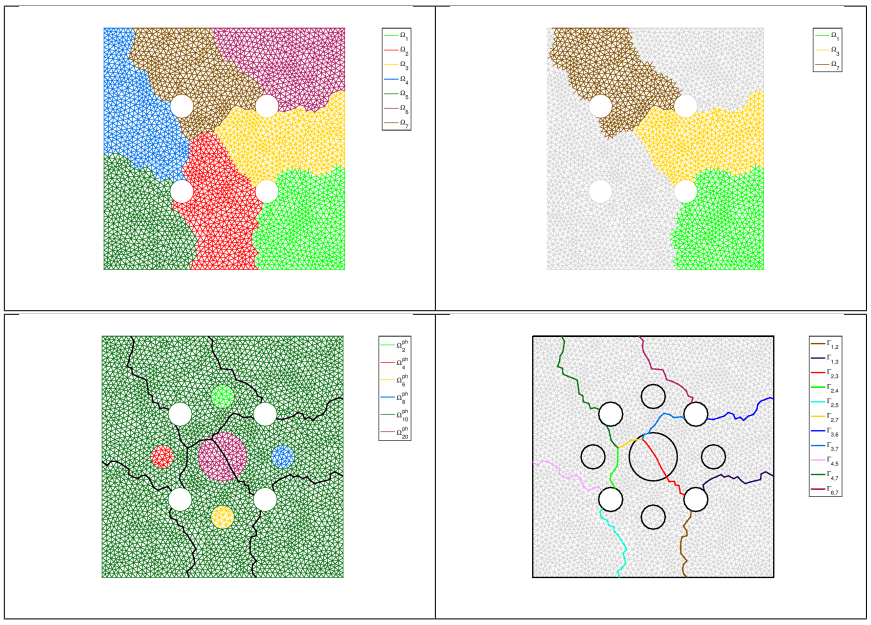
Th.plotmesh(Name,Value, ...) specifies function options using one or more Name,Value pair arguments. Options of first level are

- `'parts'` : to select the labels of the partitions to display,
- `'d'` : to select the dimensional d-simplices elements to display,
- `'labels'` : to select the labels of the elements to display,
- `'color'` : to specify the color of the displayed elements.
- `'legend'` : add a legend to graph if true (default : false)

The options of second level depend on the type of elementaries mesh elements to represent. (see plotmesh function of `mooMesh` class)

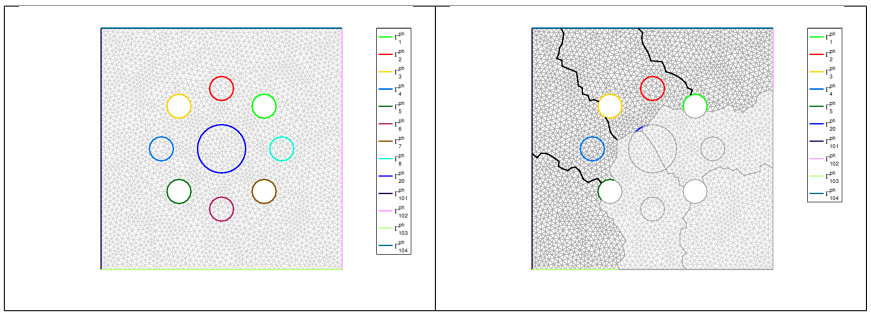
2.3.1 2D example

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



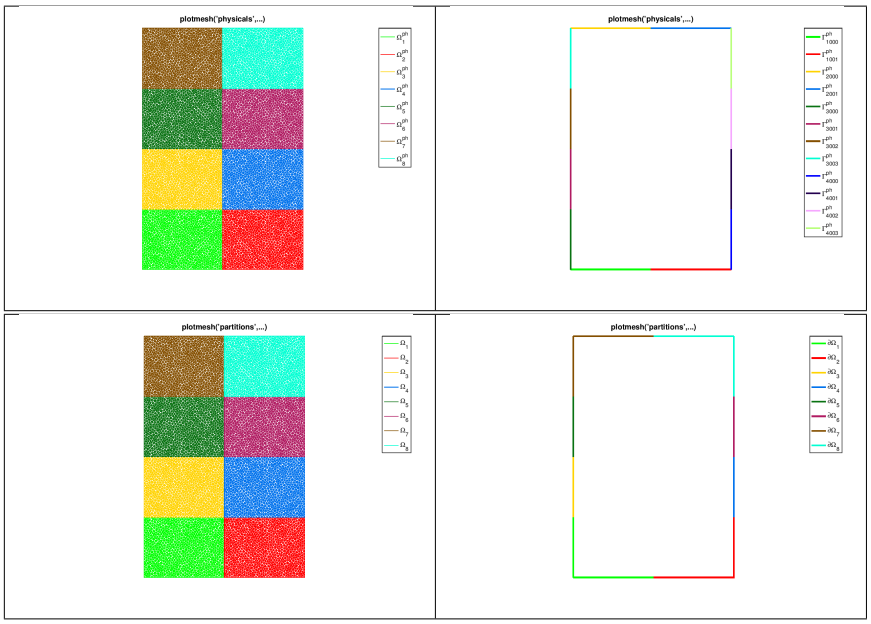
```
pTh=fc_psimesh.build_psimesh('condenser11','N',25,'np',7);
figure(1)
pTh.plotmesh('legend',true)
axis off;axis image;
figure(2)
pTh.plotmesh('parts',[1,3,7],'legend',true)
hold on;axis off;axis image
pTh.plotmesh('parts',[2,4,5,6],'color','LightGrey')
figure(3)
pTh.plotmesh('physicals','legend',true)
hold on;axis off;axis image
pTh.plotmesh('interfaces','color','black','LineWidth',1.5)
figure(4)
pTh.plotmesh('color','LightGrey')
hold on;axis off;axis image
pTh.plotmesh('physicals','d',1,'color','black', ...
    'LineWidth',1.5)
pTh.plotmesh('interfaces','legend',true,'LineWidth',1.5)
set(legend(),'location','NorthEastOutside')
```

Listing 8: Function plotmesh for 2D partitioned mesh



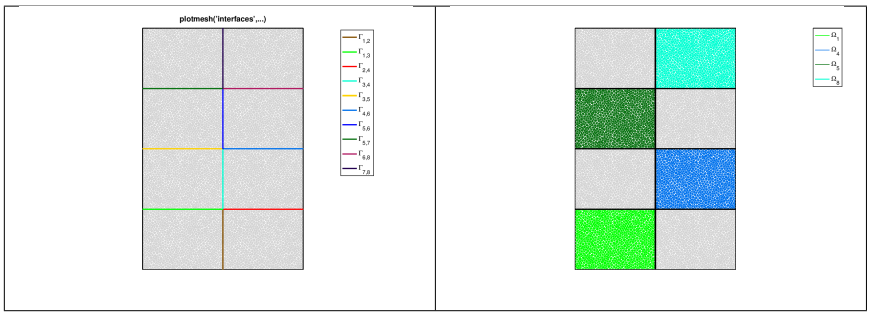
```
pTh=fc_psimesh.build_psimesh('condenser11','N',25,'np',7);
figure(1)
pTh.plotmesh('color','LightGrey')
hold on;axis off;axis image;
pTh.plotmesh('physicals','d',1,'legend',true, ...
    'LineWidth',1.5)
figure(2)
pTh.plotmesh('parts',1:3,'color','LightGrey')
hold on;axis off;axis image;
pTh.plotmesh('parts',4:7,'color','DarkGrey')
pTh.plotmesh('physicals','d',1,'parts',4:7, ...
    'legend',true, 'LineWidth',1.5)
pTh.plotmesh('interfaces','parts',4:7, ...
    'color','black','LineWidth',1.5)
pTh.plotmesh('interfaces','parts',1:3, ...
    'color','DarkGrey','LineWidth',1)
pTh.plotmesh('physicals','d',1,'parts',1:3,'color','DarkGrey', ...
    'LineWidth',1)
```

Listing 9: Function plotmesh for 2D partitioned mesh (suite)




```
meshfile=gmsh.buildpartrectangle(2,3,2,4,30);
Th=siMesh(meshfile);
pTh=psiMesh(Th);
sTitle='plotmesh(''physicals'',...)';
figure(1)
pTh.plotmesh('physicals','legend',true)
axis off;axis image;title(sTitle);
figure(2)
pTh.plotmesh('physicals','d',1,'legend',true,'LineWidth',2)
axis off;axis image;title(sTitle)
sTitle='plotmesh(''partitions'',...)';
figure(3)
pTh.plotmesh('partitions','legend',true)
axis off;axis image;title(sTitle)
figure(4)
pTh.plotmesh('partitions','d',1,'legend',true, ...
    'LineWidth',2)
axis off;axis image;title(sTitle)
```

Listing 10: Function plotmesh for 2D partitioned rectangle mesh

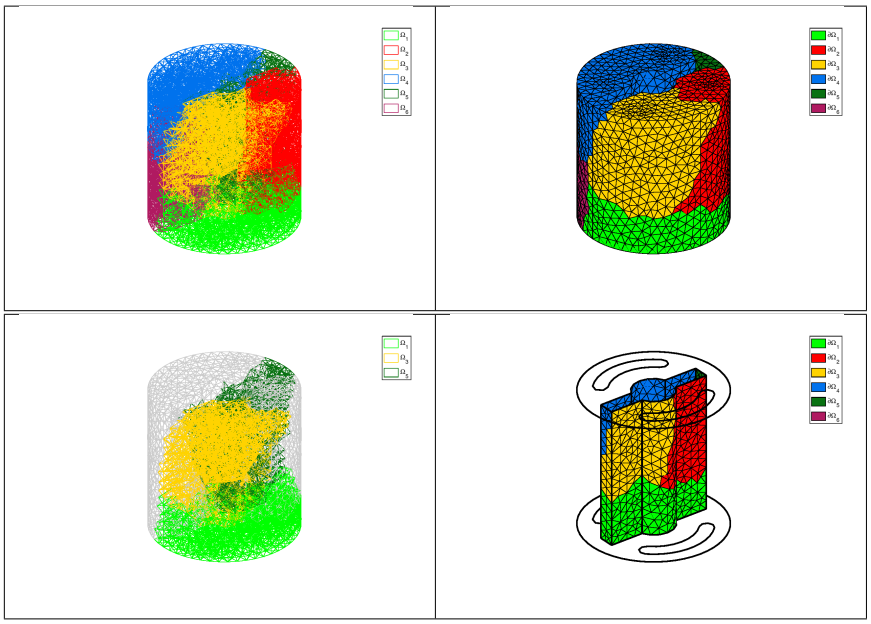



```
meshfile=gmsh.buildpartrectangle(2,3,2,4,30);
Th=siMesh(meshfile);
pTh=psiMesh(Th);
sTitle='plotmesh(''interfaces'',...)';
figure(1)
pTh.plotmesh('color','LightGrey')
hold on;axis off;axis image;
pTh.plotmesh('d',1,'color','black')
pTh.plotmesh('interfaces','legend',true,'Linewidth',1.5)
title(sTitle)
figure(2)
pTh.plotmesh('parts',[2,3,6,7],'color','LightGrey')
hold on;axis off;axis image;
pTh.plotmesh('parts',[1,4,5,8],'legend',true)
pTh.plotmesh('d',1,'color','black')
pTh.plotmesh('interfaces','color','black','Linewidth',1.5)
```

Listing 11: Function plotmesh for 2D partitioned rectangle mesh (more)

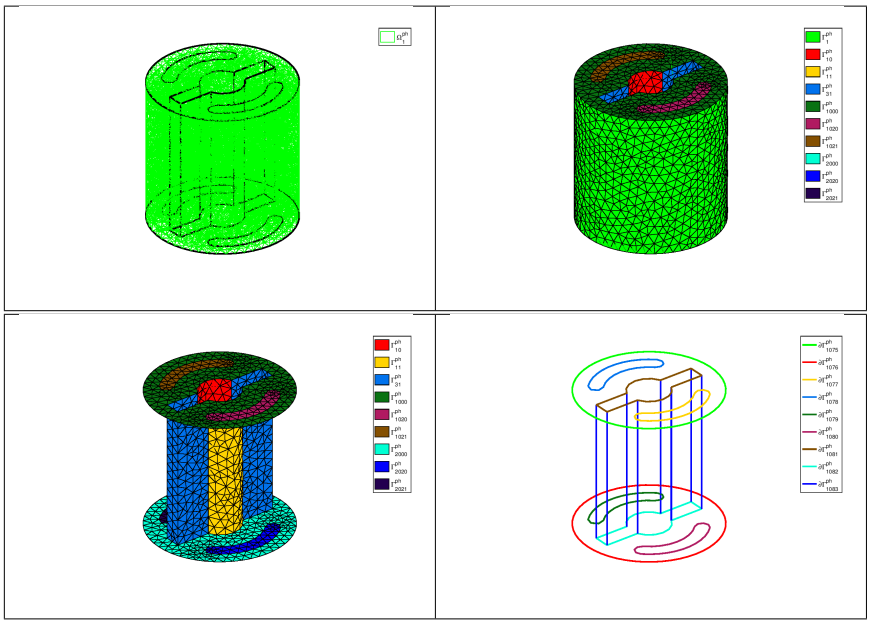
2.3.2 3D example

The following example use the *.geo* file `cylinderkey03.geo` which is in the directory `geodir/3d` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



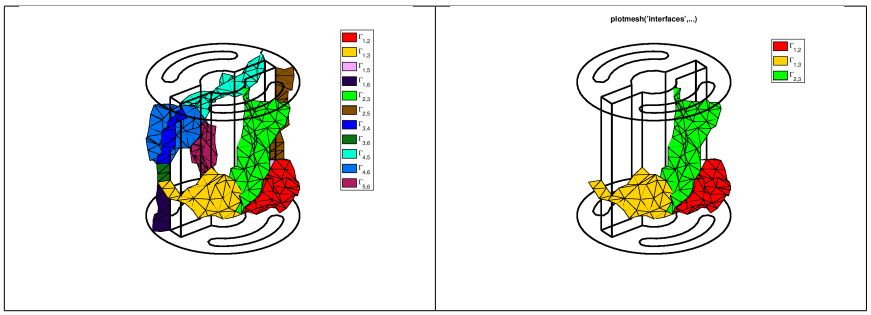
```
geofile='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
figure(1)
pTh.plotmesh('legend',true)
axis off;axis image
figure(2)
pTh.plotmesh('d',2,'legend',true)
view(3);axis off;axis image
figure(3)
pTh.plotmesh('parts',1:2:7,'legend',true)
hold on;axis off;axis image
pTh.plotmesh('d',2,'parts',2:2:6', ...
    'EdgeColor',0.8*[1,1,1], ...
    'EdgeAlpha',0.8,'FaceColor','none')
figure(4)
pTh.plotmesh('d',2,'labels',[10,11,31],'legend',true)
hold on;
pTh.plotmesh('d',1,'color','black', 'LineWidth',2);
axis off;axis image;
```

Listing 12: function lstinlineplot for a 3D partitioned mesh



```
geofile='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
figure(1)
pTh.plotmesh('physicals','legend',true)
hold on;axis off;axis image
pTh.plotmesh('d',1,'color','black', 'LineWidth',2)
figure(2)
pTh.plotmesh('physicals','d',2,'legend',true)
figure(3)
pTh.plotmesh('physicals','d',2,'legend',true, ...
    'labels',[10,11,31,1000,1020,1021,2000,2020,2021])
figure(4)
pTh.plotmesh('physicals','d',1,'legend',true, ...
    'LineWidth',2)
axis off;axis image
```

Listing 13: function lstinlineplot for a 3D partitioned mesh

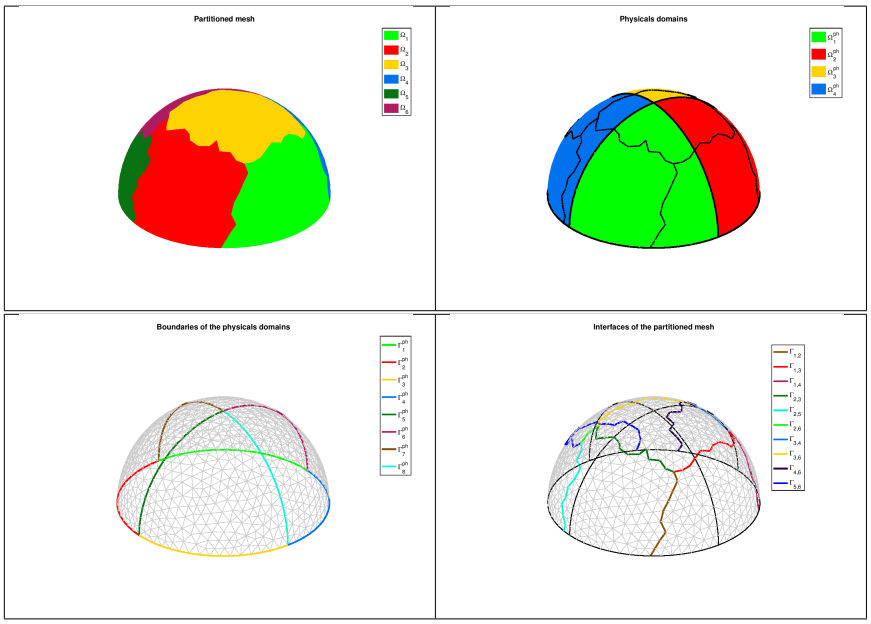



```
geofile='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
figure(1)
pTh.plotmesh('interfaces','legend',true)
hold on;axis off;axis image
pTh.plotmesh('d',1,'color','black', 'LineWidth',2)
figure(2)
pTh.plotmesh('d',1,'color','black')
axis off;axis image;
hold on
pTh.plotmesh('interfaces','ij',[1,3;2,3;1,2],'legend',true)
hold on;axis off;axis image;
pTh.plotmesh('d',1,'color','black', 'LineWidth',2)
title('plotmesh(''interfaces'',...)')
```

Listing 14: function lstinlineplot for a 3D partitioned mesh

2.3.3 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.




```
pTh=fc_psimesh.build_psimesh('demisphere4surf','N',ns*10, ...
    'np',6,'d',2);
figure(1);
pTh.plotmesh('EdgeColor','none','legend',true);
axis off;axis image
title('Partitioned_mesh')
figure(2)
pTh.plotmesh('physicals','EdgeColor','None','legend',true);
hold on
pTh.plotmesh('physicals','d',1,'color','k','LineWidth',2);
pTh.plotmesh('interfaces','color','k','LineWidth',1.5, ...
    'LineStyle','-.');
axis off;axis image
title('Physicals_domains')
figure(3)
pTh.plotmesh('FaceColor','None','EdgeColor',0.8*[1,1,1], ...
    'EdgeAlpha',0.3);
hold on
pTh.plotmesh('physicals','d',1,'LineWidth',2,'legend',true);
axis off;axis image
title('Boundaries_of_the_physicals_domains')
figure(4)
pTh.plotmesh('FaceColor','None','EdgeColor',0.8*[1,1,1], ...
    'EdgeAlpha',0.3);
hold on
pTh.plotmesh('physicals','d',1,'color','k','LineWidth',1);
pTh.plotmesh('interfaces','LineWidth',2,'legend',true);
axis off;axis image
title('Interfaces_of_the_partitioned_mesh')
```

Listing 15: function lstinlineplot for a 3D partitioned surface mesh

2.4 function PLOT

The method `plot` displays datas on the partitioned mesh or parts of the partinioned mesh defined by an `psiMesh` object.

Syntaxe With pTh a `psiMesh` object

```
pTh.plot(u)
pTh.plot(u,'interfaces')
pTh.plot(u,Name,Value, ...)
pTh.plot(u,'interfaces',Name,Value, ...)
```

Description

pTh.**plot**(u) displays data u on all the pTh.d-dimensional simplices elements. The data u is either a cell array of size pTh.np or an 1D-array of size pTh.nq or pTh.nqGlobal or pTh.nqParent.

pTh.**plot**(u,'interfaces') displays data u on all the interfaces.

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

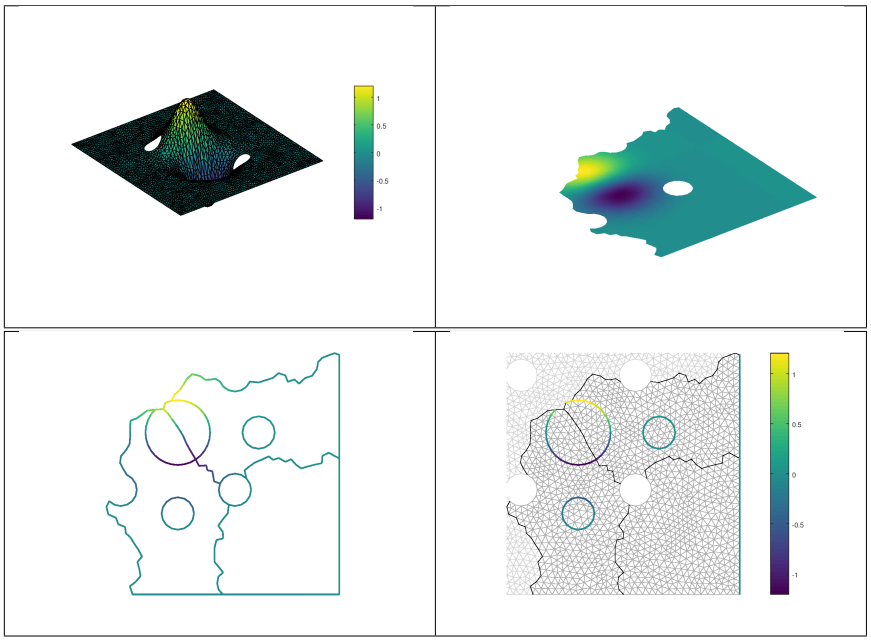
- `'parts'` : to select the labels of the partitions on which to display the data u,
- `'d'` : to specify the dimension of the simplices elements (default : Th.d)
- `'labels'` : to select the labels of the elements on which to represent data,
- `'plan'` : if true, (default : false)

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

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

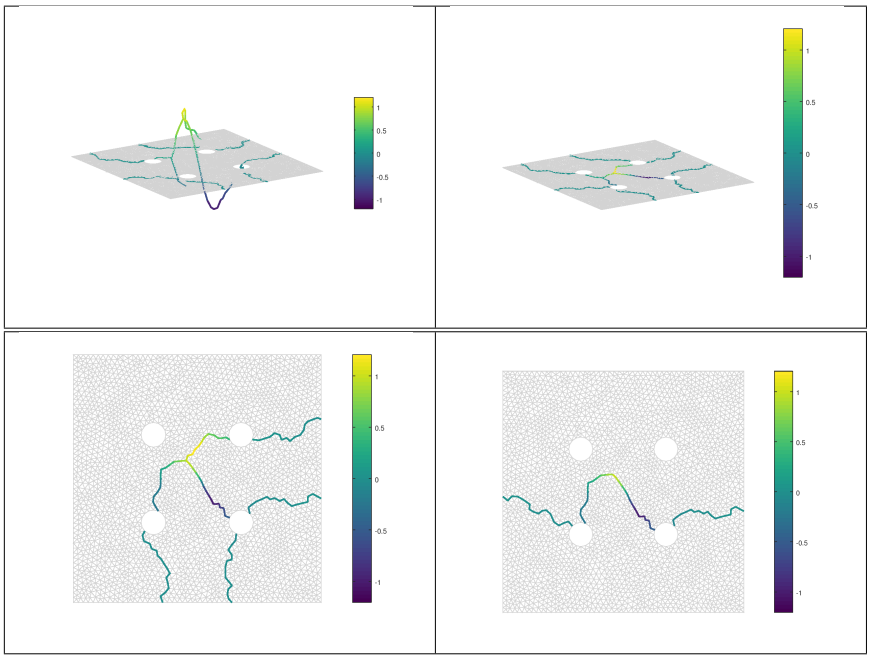
2.4.1 2D example

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



```
pTh=fc_psimesh.build_psimesh('condenser11','N',25,'np',7);
U=pTh.eval(@(x,y) 5*exp(-3*(x.^2+y.^2)).*cos(x).*sin(y));
figure(1)
pTh.plot(U)
axis image;axis off
view(3)
colorbar
figure(2)
pTh.plot(U,'parts',1:3,'EdgeColor','none', ...
    'FaceColor','interp','plan',true)
hold on;axis image;axis off
view(3)
figure(3)
pTh.plot(U,'d',1,'parts',1:3,'LineWidth',2)
hold on;axis image;axis off
figure(4)
pTh.plotmesh('parts',1:3,'color','DarkGrey')
hold on;axis image;axis off
pTh.plotmesh('parts',4:7,'color','LightGrey')
pTh.plotmesh('interfaces','parts',1:3,'color','black')
pTh.plot(U,'d',1,'parts',1:3,'labels',[20,2:2:8,101:104], ...
    'LineWidth',2,'plan',true)
colorbar
```

Listing 16: 2D partitioned mesh : **plot** function

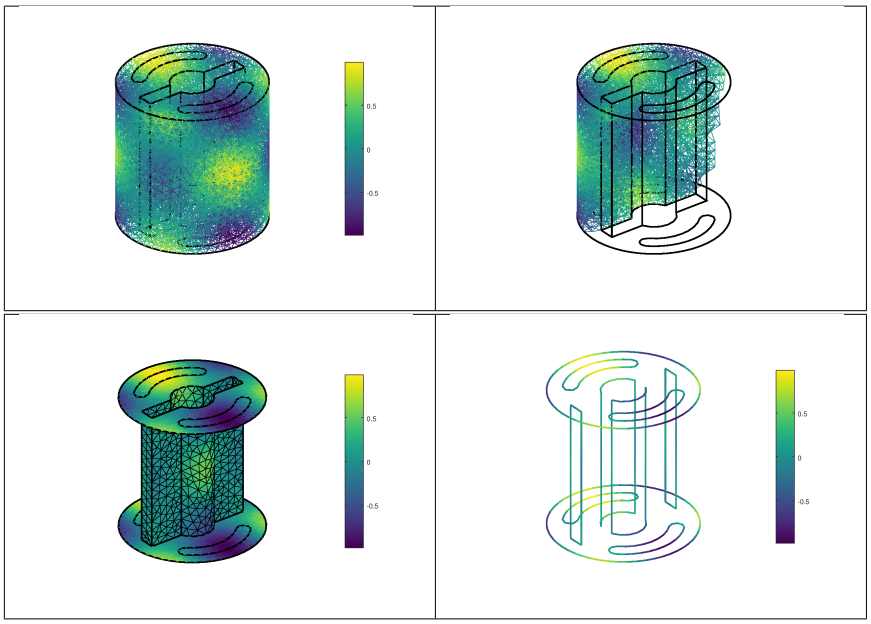


```
pTh=fc_psimesh.build_psimesh('condenser11','N',25,'np',7);
U=pTh.eval(@(x,y) 5*exp(-3*(x.^2+y.^2)).*cos(x).*sin(y));
figure(1)
pTh.plotmesh('color','LightGrey')
hold on;axis image;axis off
pTh.plot(U,'interfaces','Linewidth',2)
view([-33,16])
colorbar
figure(2)
pTh.plotmesh('color','LightGrey')
hold on;axis image;axis off
pTh.plot(U,'interfaces','plan',true,'Linewidth',2)
colorbar
view([-33,16])

figure(3)
pTh.plotmesh('color','LightGrey')
hold on;axis image;axis off
pTh.plot(U,'interfaces','parts',1:3,'LineWidth',2, ...
    'plan',true)
colorbar

figure(4)
pTh.plotmesh('color','LightGrey')
hold on;axis image;axis off
pTh.plot(U,'interfaces','ij',[4,5;2,7;2,3;2,4;1,3], ...
    'plan',true,'LineWidth',2)
colorbar
```

Listing 17: 2D partitioned mesh : **plot** function on interfaces

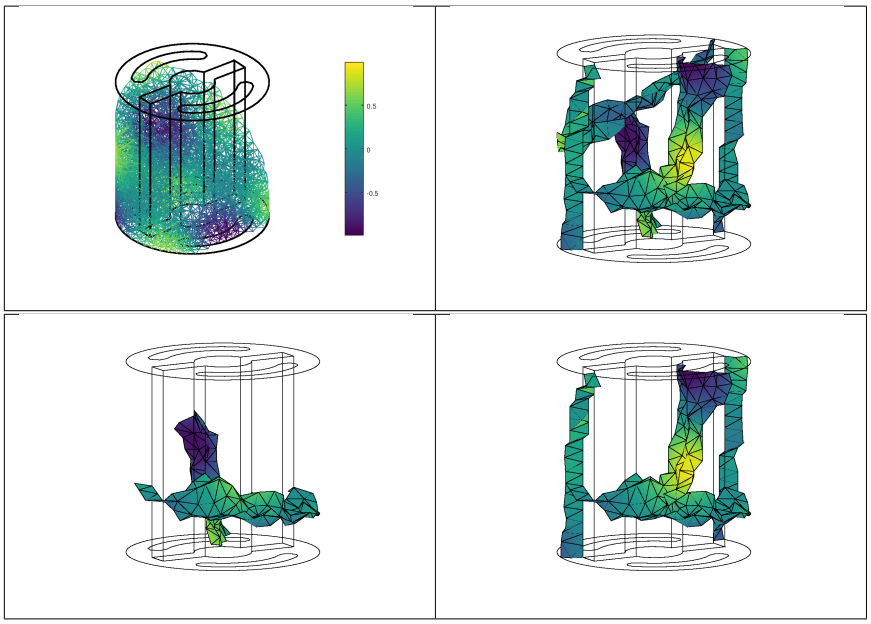
2.4.2 3D example

The following example use the *.geo* file `cylinderkey03.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='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
u=@(x,y,z) cos(3*x).*sin(2*y).*cos(pi*z);
U=pTh.eval(u);
figure(1)
pTh.plot(U)
hold on
pTh.plotmesh('d',1,'color','black','Linewidth',2)
axis off;axis image;
colorbar
figure(2)
pTh.plot(U,'parts',4:6)
hold on
pTh.plotmesh('d',1,'color','black','Linewidth',2)
axis off;axis image;
figure(3)
pTh.plot(U,'d',2,'labels',[1000 1020 1021 2000 2020 ...
    2021],'EdgeColor','interp','FaceColor','interp')
axis off;axis image;hold on
pTh.plot(U,'d',2,'labels',[10 11 31])
pTh.plotmesh('d',1,'color','black','Linewidth',2)
colorbar
figure(4)
pTh.plot(U,'d',1,'linewidth',2)
axis off;axis image;
colorbar
```

Listing 18: 3D partitioned mesh : **plot** function

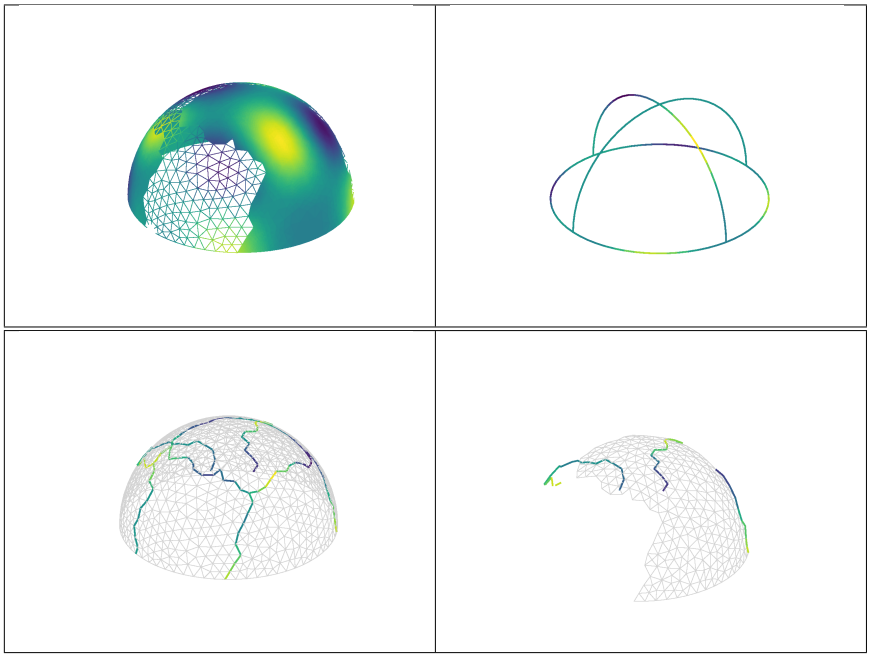


```
geofile='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
u=@(x,y,z) cos(3*x).*sin(2*y).*cos(pi*z);
U=pTh.eval(u);
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 -1 1]);
figure(1)
pTh.plot(U,'cutPlan',P)
axis off;hold on
pTh.plotmesh('d',1,'color','black','Linewidth',2)
axis image;colorbar
figure(2)
pTh.plotmesh('d',1,'color','black')
axis off;axis image;hold on
pTh.plot(U,'interfaces')
view(-30,11)
figure(3)
pTh.plotmesh('d',1,'color','black')
axis off;axis image;hold on
pTh.plot(U,'interfaces','ij',[1,2;1,3;5,6])
view(-30,11)
figure(4)
pTh.plotmesh('d',1,'color','black')
axis off;axis image;hold on
pTh.plot(U,'interfaces','parts',1:3)
view(-30,11)
```

Listing 19: 3D partitioned mesh : **plot** function (more)

2.4.3 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.



```
pTh=fc_psimesh.build_psimesh('demisphere4surf','N',ns*10, ...
    'np',6,'d',2);
u=@(x,y,z) cos(5*x).*sin(3*y).*cos(4*z);
U=pTh.eval(u);
figure(1);
pTh.plot(U,'FaceColor','interp','EdgeColor','none', ...
    'parts',1:2:6)
hold on;axis off;axis image
pTh.plot(U,'FaceColor','none','EdgeColor','interp', ...
    'parts',2:2:6)
figure(2);
pTh.plot(U,'d',1,'LineWidth',2,'labels',1:8)
hold on;axis off;axis image
figure(3);
pTh.plotmesh('FaceColor','none','Edgecolor','LightGrey')
hold on;axis off;axis image
pTh.plot(U,'interfaces','LineWidth',2)
figure(4);
pTh.plotmesh('parts',[1,3],'FaceColor','none', ...
    'Edgecolor','LightGrey')
hold on;axis off;axis image
pTh.plot(U,'interfaces','ij',[4,1;4,6;5,6],'LineWidth',2)
```

Listing 20: 3D partitioned surface mesh : **plot** function

2.4.4 function PLOTISO

The method `plotIsolines` displays isolines from datas on the partitioned mesh or parts of the partitioned mesh defined by an `psiMesh` object. This function only works with 2-simplices in space dimension 2 or 3.

Syntaxe

```
pTh.plotiso(u)
pTh.plotiso(u,Name,Value, ...)
```

Description

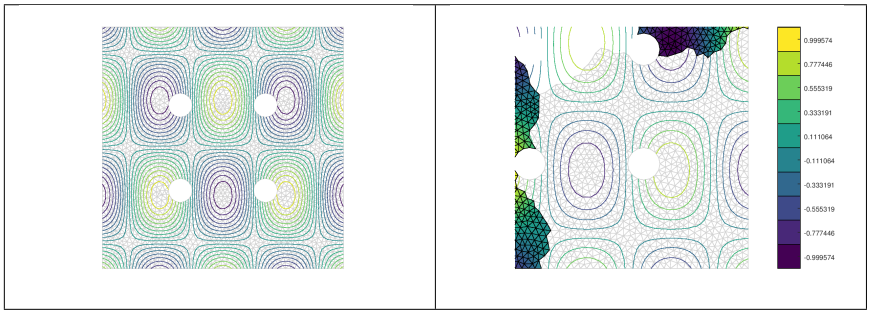
See method plotiso of `siMesh` class for description (see **??**).

A supplementary Name,Value pair was added :

- `'parts'` : to select the labels of the partitions on which to display isolines.

2.4.5 2D example

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

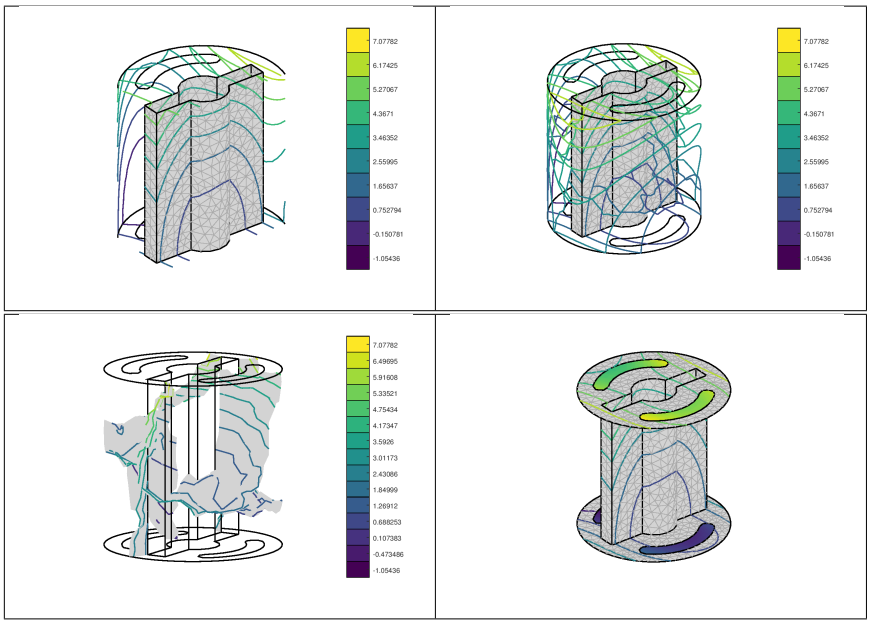



```
pTh=fc_psimesh.build_psimesh('condenser11','N',25,'np',7, ...
    'verbose',2,'physlab',true);
U=pTh.eval(@(x,y) cos(3*x).*sin(2*y));
figure(1)
pTh.plotmesh('Color','LightGrey')
hold on;axis image;axis off
pTh.plotiso(U,'plan',true,'niso',20)
axis off;axis image;
figure(2)
pTh.plotmesh('parts',1:3,'Color','LightGrey')
hold on;axis image;axis off
pTh.plotiso(U,'parts',[1:3,7],'plan',true,'isocolorbar',true)
pTh.plot(U,'parts',4:6,'plan',true)
```

Listing 21: 2D partitioned mesh : plotiso function

2.4.6 3D example

The following example use the *.geo* file `cylinderkey03.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.

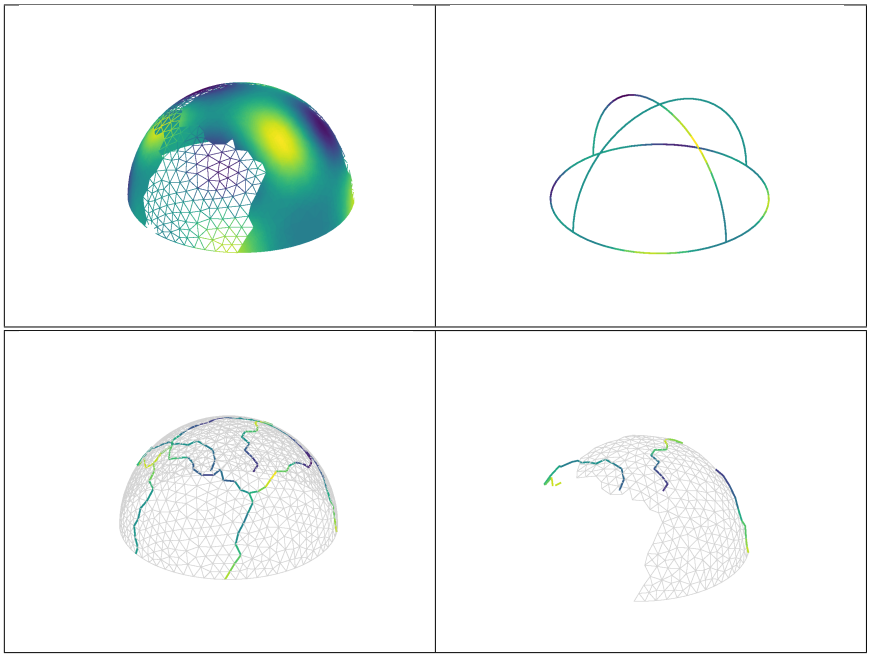


```
pTh=fc_psimesh.build_psimesh('cylinderkey03','N',ns*8, ...
    'np',6,'geodir',geodir3d,'d',3,'verbose',2);
U=pTh.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
pTh.plotmesh('d',2,'EdgeColor','DarkGrey', ...
    'faceColor','LightGrey','labels',[10,11,31])
hold on;axis image;axis off
pTh.plotmesh('d',1,'Color','black','LineWidth',1.5);
pTh.plotiso(U,'isocolorbar',true,'LineWidth',1.5)
figure(2)
pTh.plotmesh('d',2,'EdgeColor','DarkGrey', ...
    'faceColor','LightGrey','labels',[10,11,31])
hold on;
pTh.plotmesh('d',1,'Color','black','LineWidth',1.5);
pTh.plotiso(U,'interfaces',true,'isocolorbar',true, ...
    'LineWidth',1.5)
axis image;axis off
figure(3)
pTh.plotmesh('interfaces','EdgeColor','none', ...
    'faceColor','LightGrey')
hold on;axis image;axis off
pTh.plotmesh('d',1,'Color','black','LineWidth',1.5);
pTh.plotiso(U,'interfaces',true,'physicals',false, ...
    'niso',15,'isocolorbar',true,'LineWidth',1.5)
view(-58,11)
figure(4)
pTh.plotmesh('d',2,'EdgeColor','DarkGrey', ...
    'faceColor','LightGrey','labels',[10,11,31,1000,2000])
hold on;axis image;axis off
pTh.plotmesh('d',1,'Color','black','LineWidth',1.5);
pTh.plotiso(U,'labels',[10,11,31,1000,2000],'LineWidth',1.5)
pTh.plot(U,'d',2,'labels',[1020,1021,2020,2021], ...
    'LineWidth',1.5,'Edgecolor','none','FaceColor','interp')
```

Listing 22: 3D partitioned mesh : plotiso function

2.4.7 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.



```
pTh=fc_psimesh.build_psimesh('demisphere4surf','N',ns*10, ...
    'np',6,'d',2);
u=@(x,y,z) cos(5*x).*sin(3*y).*cos(4*z);
U=pTh.eval(u);
figure(1)
pTh.plotmesh('parts',[1,5,6],'EdgeColor','DarkGrey', ...
    'faceColor','LightGrey')
hold on;axis off;axis image;
pTh.plotiso(U,'parts',[1,5,6],'LineWidth',2)
pTh.plotiso(U,'parts',[2:4],'LineWidth',1)
figure(2)
pTh.plot(U,'parts',2:4,'FaceColor','interp', ...
    'EdgeColor','none')
hold on;axis off;axis image;
pTh.plotmesh('parts',[1,5,6],'EdgeColor','DarkGrey', ...
    'faceColor','LightGrey')
pTh.plotiso(U,'parts',[1,5,6],'LineWidth',1.5)
```

Listing 23: 3D partitioned surface mesh : plotiso function

2.4.8 function SLICEMESH

The method `slicemesh` displays intersection of a plane and a 3D partitioned mesh defined by an `psiMesh` object.

Syntaxe

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

Description

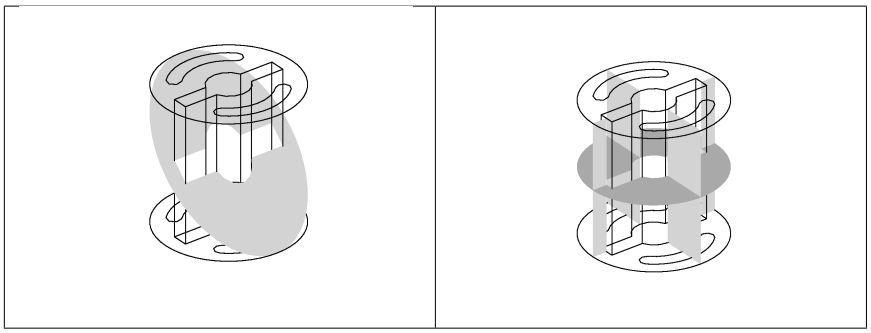
See method slicemesh of `siMesh` class for description (see ??).

A supplementary Name,Value pair was added :

- `'parts'` : to select the labels of the partitions on which to display the slice.

2.4.9 3D example

The following example use the *.geo* file `cylinderkey03.geo` which is in the directory `geodir/3d` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
geofile='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
U=pTh.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
pTh.plotmesh('d',1,'LineWidth',1,'color','k')
hold on
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 -1 1]);
pTh.slicemesh(P,'Color','LightGrey')
axis off;axis image;

figure(2)
pTh.plotmesh('d',1,'LineWidth',1,'color','k')
hold on
P=fc_tools.graphics.PlaneCoefs([0 0 1],[-1 0 0]);
pTh.slicemesh(P,'Color','LightGrey')
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 -1 0]);
pTh.slicemesh(P,'Color','LightGrey')
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 0 -1]);
pTh.slicemesh(P,'Color','DarkGrey')
axis off;axis image;
```

Listing 24: function lstinlineslicemesh for a 3D partitioned mesh

2.4.10 function SLICE

The method `slice` displays datas on the intersection of a plane and a 3D partitioned mesh defined by an `psiMesh` object.

Syntaxe

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

Description

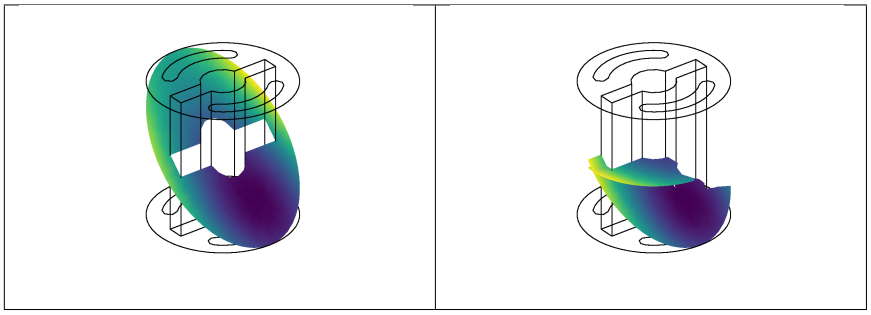
See method **slice** of `siMesh` class for description (see **??**).

A supplementary Name,Value pair was added :

- `'parts'` : to select the labels of the partitions on which to display the slice.

2.4.11 3D example

The following example use the *.geo* file `cylinderkey03.geo` which is in the directory `geodir/3d` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
geofile='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
U=pTh.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
pTh.plotmesh('d',1,'LineWidth',1,'color','k')
hold on
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 -1 1]);
pTh.slice(U,P)
axis off;axis image;
figure(2)
pTh.plotmesh('d',1,'LineWidth',1,'color','k')
hold on
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 -1 1]);
pTh.slice(U,P,'parts',[1,3])
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 0 -1]);
pTh.slice(U,P,'parts',[1,3])
axis off;axis image;
```

Listing 25: function lstinlineslice for a 3D partitioned mesh

2.4.12 function SLICEISO

The method `sliceiso` displays datas as isolines on the intersection of a plane and a 3D partitioned mesh defined by an `psiMesh` object.

Syntaxe

```
Th.isoslice(u,P)
Th.isoslice(u,P,Name,Value, ...)
```

Description

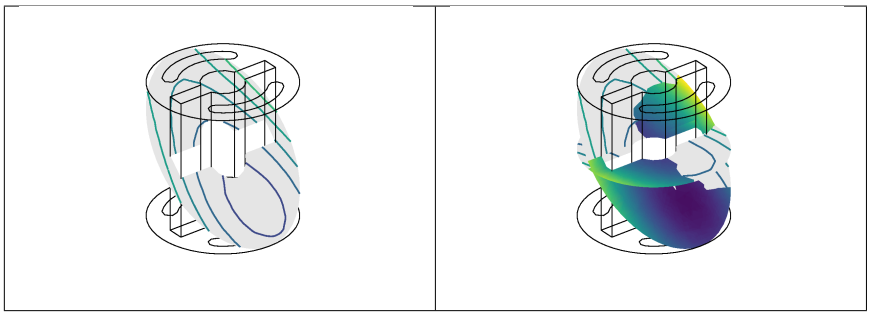
See method sliceiso of `siMesh` class for description (see **??**).

A supplementary Name,Value pair was added :

- `'parts'` : to select the labels of the partitions on which to display the slice.

2.4.13 3D example

The following example use the *.geo* file `cylinderkey03.geo` which is in the directory `geodir/3d` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
geofile='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
U=pTh.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
pTh.plotmesh('d',1,'LineWidth',1,'color','k')
hold on;axis off;axis image;
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 -1 1]);
pTh.slicemesh(P)
pTh.sliceiso(U,P,'LineWidth',2)
figure(2)
pTh.plotmesh('d',1,'LineWidth',1,'color','k')
hold on;axis off;axis image;
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 -1 1]);
pTh.slice(U,P,'parts',[1,3,5])
pTh.slicemesh(P,'parts',[2,4,6])
pTh.sliceiso(U,P,'parts',[2,4,6],'LineWidth',2)
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 0 -1]);
pTh.slice(U,P,'parts',[1,3,5])
pTh.slicemesh(P,'parts',[2,4,6])
pTh.sliceiso(U,P,'parts',[2,4,6],'LineWidth',2)
```

Listing 26: function lstinlinesliceiso for a 3D partitioned mesh

2.4.14 function plotVectorField

The method `plotVectorField` displays vector field datas on the mesh or parts of the partitioned mesh defined by an `psiMesh` object.

Syntaxe

```
pTh.plotVectorField(V)
pTh.plotVectorField(V,Name,Value, ...)
```

Description

pTh.plotVectorField(V) displays the vector field V on the mesh. The vector field V could be :

- a pTh.nq-by-pTh.dim array.
- a pTh.np-by-1 cells array where the i-th cell is an pTh.Th{i}.nq-by-pTh.dim array.

See method plotVectorField of `siMesh` class for description (see **??**).

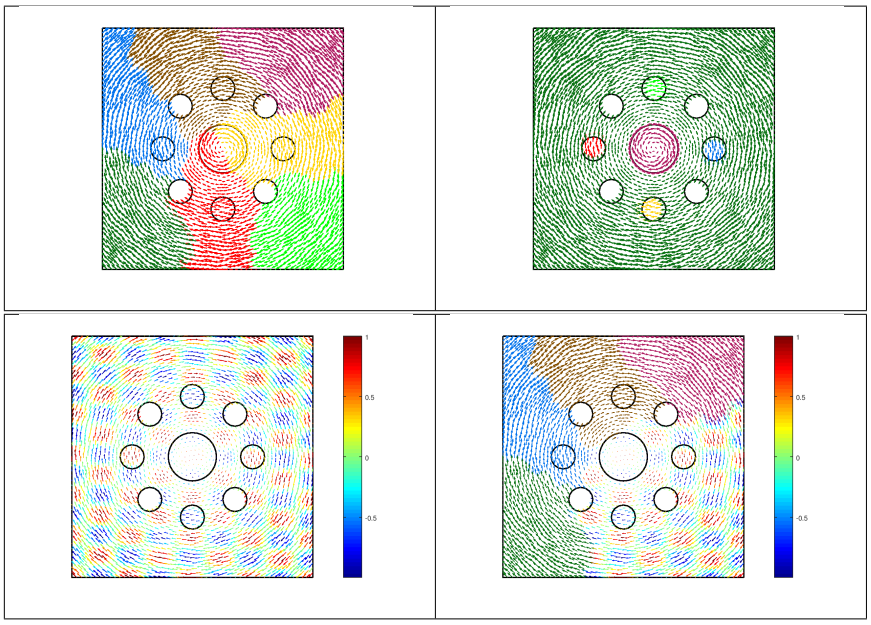
A supplementary Name,Value pair was added :

- `'parts'` : to select the labels of the partitions on which to display isolines.
- `'physicals'` : if set to true, the vector field is colorized with the colors of the physical meshes. Default is false and the vector field is colorized with the colors of the partition meshes.

The `'colordata'` option accept also, as value, a cell array of size pTh.np.

2.4.15 2D example

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

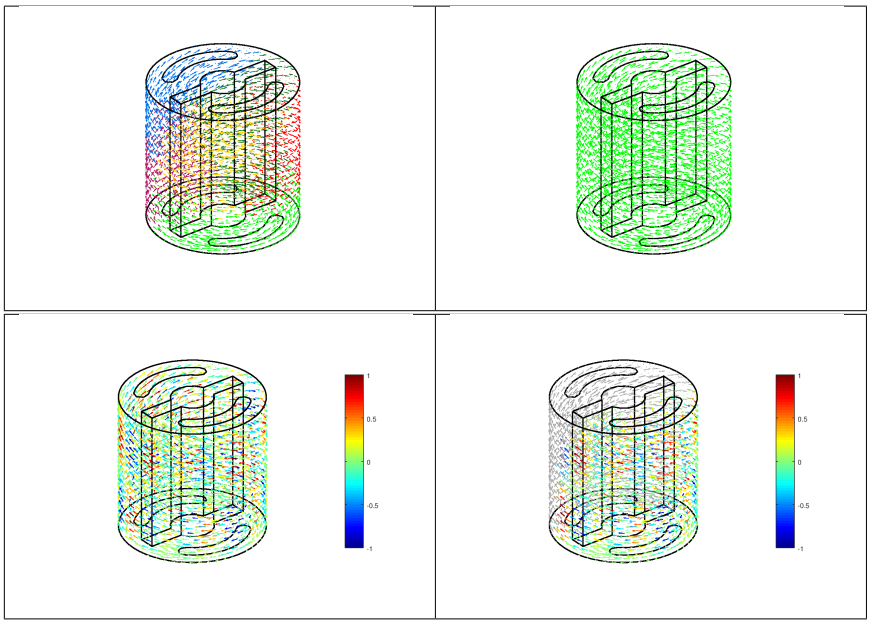


```
pTh=fc_psimesh.build_psimesh('condenser11','N',25,'np',7);
u=@(x,y) cos(2*pi*x).*cos(3*pi*y);;
U=pTh.eval(u);
w={@(x,y) y.*cos(-(x.^2+y.^2)/10);@(x,y) ...
    -x.*cos(-(x.^2+y.^2)/10)};
W=pTh.eval(w);
figure(1)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'LineWidth',1)
figure(2)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'Physicals',true,'LineWidth',1)
figure(3)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'colordata',U, 'scale',0.05)
colormap('jet');colorbar
figure(4)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'parts',1:3,'colordata',U, ...
    'scale',0.05)
pTh.plotVectorField(W,'parts',4:7, 'scale',0.05)
colormap('jet');colorbar
```

Listing 27: 2D partitioned mesh : plotVectorField function

2.4.16 3D example

The following example use the *.geo* file `cylinderkey03.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.

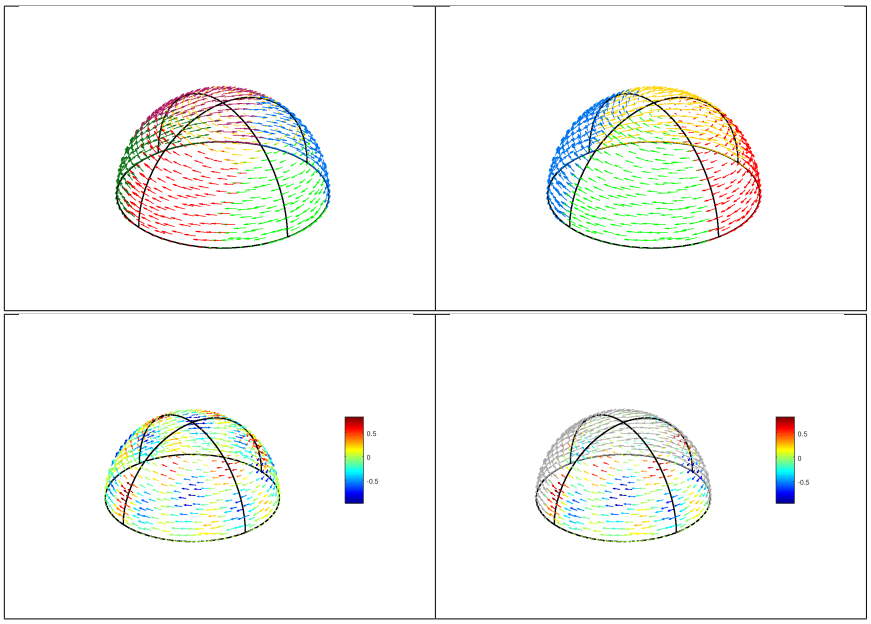


```
geofile='cylinderkey03';
pTh=fc_psimesh.build_psimesh(geofile,'N',ns*8,'np',6, ...
    'geodir',geodir3d,'d',3,'verbose',2);
u=@(x,y,z) cos(2*pi*x).*cos(3*pi*y).*sin(pi*z);
U=pTh.eval(u);
w={@(x,y,z) y.*cos(-(x.^2+y.^2)/10);@(x,y,z) ...
    -x.*cos(-(x.^2+y.^2)/10);@(x,y,z) z/5 };
W=pTh.eval(w);
figure(1)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W)
figure(2)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'Physicals',true)
figure(3)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'colordata',U, 'scale',0.1)
colormap('jet');colorbar
figure(4)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'parts',1:3,'colordata',U, ...
    'scale',0.1)
pTh.plotVectorField(W,'parts',4:6, ...
    'scale',0.1,'color','DarkGrey')
colormap('jet');colorbar
```

Listing 28: 3D partitioned mesh : plotVectorField function

2.4.17 3D surface example

The following example use the *.geo* file `demisphere4surf.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.



```
pTh=fc_psimesh.build_psimesh('demisphere4surf','N',ns*10, ...
    'np',6,'d',2);
u=@(x,y,z) cos(2*pi*x).*cos(3*pi*y).*sin(pi*z);
U=pTh.eval(u);
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=pTh.eval(w);
figure(1)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W)
figure(2)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'Physicals',true)
figure(3)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'colordata',U, 'scale',0.1)
colormap('jet');colorbar
figure(4)
pTh.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on;axis off;axis image
pTh.plotVectorField(W,'parts',1:3,'colordata',U, ...
    'scale',0.1)
pTh.plotVectorField(W,'parts',4:6, ...
    'scale',0.1,'color','DarkGrey')
colormap('jet');colorbar
```

Listing 29: 3D partitioned surface mesh : plotVectorField function