FC-SIMESH Matlab toolbox, User's Guide *

François Cuvelier†

February 5, 2017

Abstract

This object-oriented Matlab toolbox allows to use simplices meshes generated from `gmsh` (in dimension 2 or 3) or an hypercube triangulation (in any dimension). A particular care was taken to the graphics representations of meshes and datas on meshes.

Contents

*Compiled with Matlab 2015b

†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.

1 Mesh Objects

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

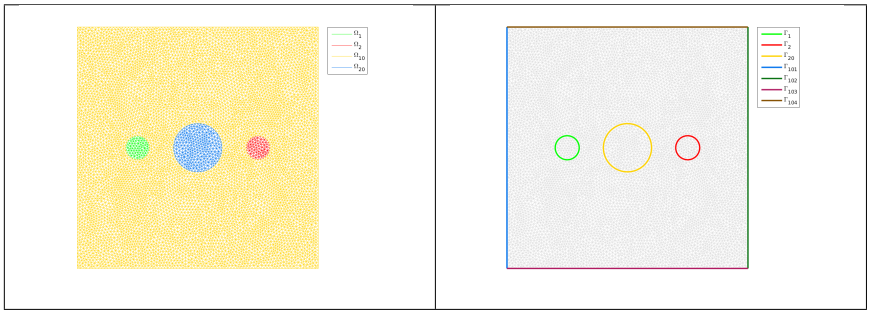
$$K = \left\{ \sum_{i=0}^{k} \lambda_i \mathrm{q}^i \mid \lambda_i \geqslant 0,\ i \in [\![0, k]\!],\ \text{with} \sum_{i=0}^{k} \lambda_i = 1 \right\}.$$

We denote by **k-simplicial elementary mesh** in $\mathbb{R}^{\dim}$, $k \leqslant \dim$, a mesh with **unique label** only composed with k-simplices.

A d-**simplicial mesh** in $\mathbb{R}^{\dim}$, $\mathrm{d} \leqslant \dim$, is an union of k-simplicial elementary meshes with $k \in [\![0, \mathrm{d}]\!]$.

1.1 Mesh samples

1.1.1 2-simplicial mesh in $\mathbb{R}^2$



```
meshfile=gmsh.buildmesh2d('sample20',20,'force',false);
Th=siMesh(meshfile);
figure(1)
Th.plotmesh('legend',true)
axis off;axis image
set(legend(),'Location','NorthEastOutside')
figure(2)
Th.plotmesh('color',[0.85,0.85,0.85],'legend',false)
hold on
Th.plotmesh('d',1,'legend',true,'Linewidth',2 );
set(legend(),'Location','NorthEastOutside')
axis off;axis image
```

Listing 1: Mesh from `sample20.geo`, label of the domains (left) and label of the boundaries (right)

For example, from figures of Listing 13 the complete domain is

$$\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_{10} \cup \Omega_{20}$$

and we note

$$\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_{20} \cup \Gamma_{101} \cup \Gamma_{102} \cup \Gamma_{103} \cup \Gamma_{104}.$$

So this mesh is 2-simplicial mesh in $\mathbb{R}^2$ and is composed of :

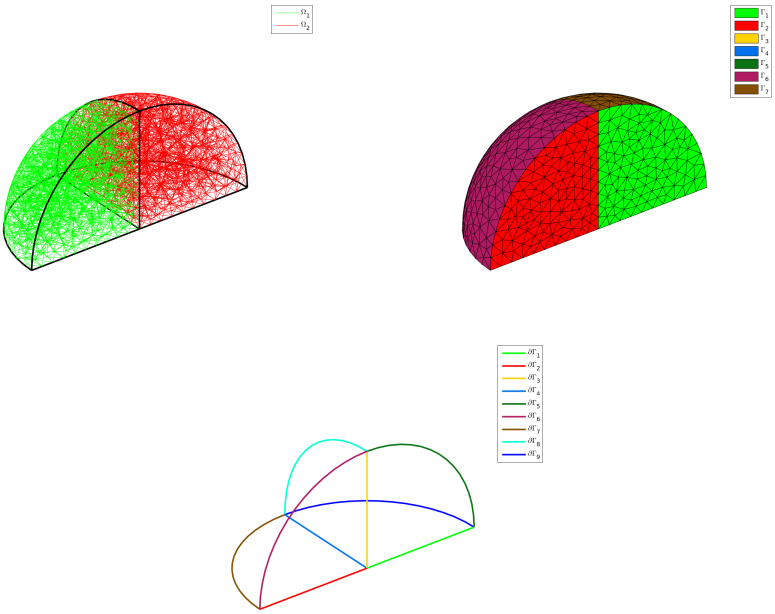
- four 2-simplicial elementary meshes : Ω_i, $\forall i \in \{1, 2, 10, 20\}$
- seven 1-simplicial elementary meshes : Γ_i $\forall i \in \{1, 2, 20, 101, 102, 104\}$

1.1.2 Sample of a 3-simplicial mesh in $\mathbb{R}^3$

```
meshfile=gmsh.buildmesh3d('quart_sphere2',5);
Th=siMesh(meshfile);
figure(1);
Th.plotmesh('FaceAlpha',0.5,'legend',true);
hold on
Th.plotmesh('d',1,'color',[0,0,0],'LineWidth',2);
```

```
axis off;axis image;
figure(2);
Th.plotmesh('d',2,'legend',true);
axis off;axis image
figure(3);
Th.plotmesh('d',1,'LineWidth',2,'legend',true);
axis off;axis image
```

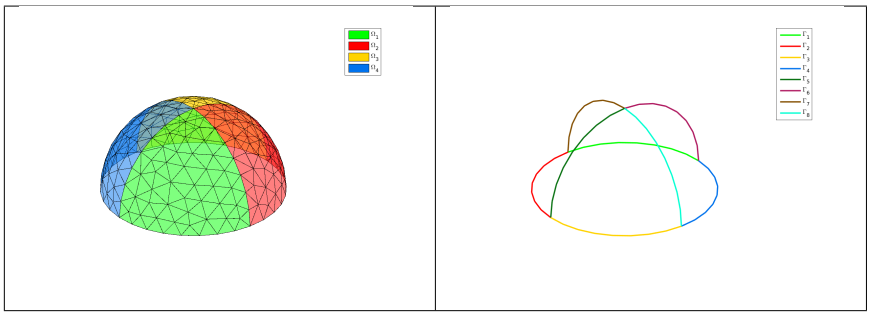
Listing 2: Mesh from `quart_sphere2.geo`



The mesh obtained from Listing 14 is a 3-simplicial mesh in $\mathbb{R}^3$ and is composed of :

- two 3-simplicial elementary meshes : Ω_i, $\forall i \in \{1, 2\}$
- seven 2-simplicial elementary meshes : Γ_i $\forall i \in [\![1, 7]\!]$
- nine 1-simplicial elementary meshes : $\partial\Gamma_i$ $\forall i \in [\![1, 9]\!]$

1.1.3 Sample of a 2-simplicial mesh in $\mathbb{R}^3$



```
meshfile=gmsh.buildmesh3ds('demisphere4surf',5,'force',true);
Th=siMesh(meshfile);
figure(1);
Th.plotmesh('FaceAlpha',0.5,'legend',true)
axis off;axis image
figure(2)
Th.plotmesh('d',1,'Linewidth',2,'legend',true);
axis off;axis image
```

Listing 3: Mesh from `demisphere4surf.geo`, label of the domains (left) and label of the boundaries (right)

The mesh obtained from Listing 9 is a 2-simplicial mesh in $\mathbb{R}^3$ and is composed of :

- four 2-simplicial elementary meshes : Ω_i, $\forall i \in [\![1,4]\!]$
- eight 1-simplicial elementary meshes : Γ_i $\forall i \in [\![1,8]\!]$

1.2 siMeshElt object

An elementary d-simplicial mesh in dimension dim is represented by the class siMeshElt . We give properties of this class :

Properties of siMeshElt object for d-simplicial elementary meshes in $\mathbb{R}^{\text{dim}}$

dim	:	integer space dimension
d	:	integer ($0 \leqslant \text{d} \leqslant \text{dim}$)
n_q	:	integer number of vertices
n_me	:	integer number of elements (d-simplices)
q	:	dim-by-n_q array of reals array of vertex coordinates
me	:	$(\text{d}+1)$-by-n_me array of integers connectivity array for **mesh elements**
vols	:	1-by-n_me array of reals array of mesh element volumes
h	:	double mesh step size (=maximum edge length in the mesh)
`toGlobal`	:	1-by-n_q array of integers convert from local to global mesh vertices numbering
`toParent`	:	1-by-n_q array of integers convert from local to parent mesh vertices numbering (same as global if not part of a partitioned mesh)

More precisely

- $\text{q}(\nu, j)$ is the ν-th coordinate of the j-th vertex, $\nu \in \{1, \ldots, \text{dim}\}$, $j \in \{1, \ldots, \text{n}_\text{q}\}$. The j-th vertex will be also denoted by $\text{q}^j = \text{q}(:, j)$.
- $\text{me}(\beta, k)$ is the storage index of the β-th vertex of the k-th element (d-simplex), in the array q, for $\beta \in \{1, ..., \text{d}+1\}$ and $k \in \{1, \ldots, \text{n}_\text{me}\}$. So $\text{q}(:, \text{me}(\beta, k))$ represents the coordinates of the β-th vertex of the k-th mesh element.
- $\text{vols}(k)$ is the volume of the k-th d-simplex .

1.3 siMesh object

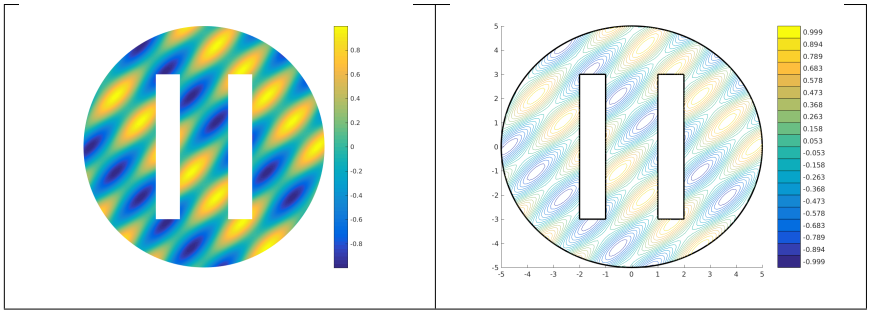
A d-simplicial mesh in dimension dim, represented as an siMesh object, is an union of siMeshElt objects which are elementary l-simplicial meshes ($l \leqslant \text{d}$) in space dimension dim.

Mesh structure associated to $\mathcal{T}_h$

dim	:	integer space dimension
d	:	integer d-dimensional simplicial mesh
`sTh`	:	array of siMeshElt objects
`nsTh`	:	number of siMeshElt objects
`sThsimp`	:	array of `nsTh` integers i-th siMeshElt object in `sTh` is a `sThsimp`(i)-simplicial elementary mesh
`sThlab`	:	array of `nsTh` integers in `sTh` label of i-th siMeshElt object in `sTh` is number `sThlab`(i)
n_q	:	integer number of vertices in $\mathcal{T}_h$
`toGlobal`	:	1-by-n_q array of integers convert from local to global mesh vertices numbering
`toParent`	:	1-by-n_q array of integers convert from local to parent mesh vertices numbering (same as global if not part of a partitioned mesh)

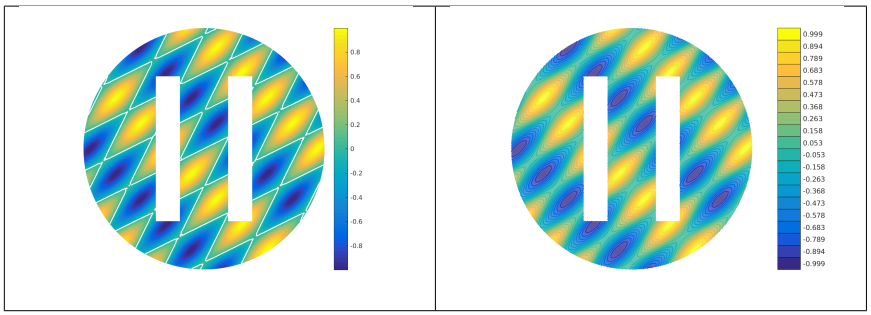
2 Data representation on meshes

2.1 Data 2D mesh



```
meshfile=gmsh.buildmesh2d('condenser',10);
Th=siMesh(meshfile);
u=@(x,y) cos(2*x-y).*sin(x-2*y);
U=Th.eval(u);
figure(1);
Th.plot(U)
shading interp
axis off;axis image
colorbar
figure(2)
axis off;axis image
Th.plotmesh('d',1,'color','k','Linewidth',2);
hold on
Th.plotiso(U,'niso',20,'isocolorbar',true,'format','%.3f');
```

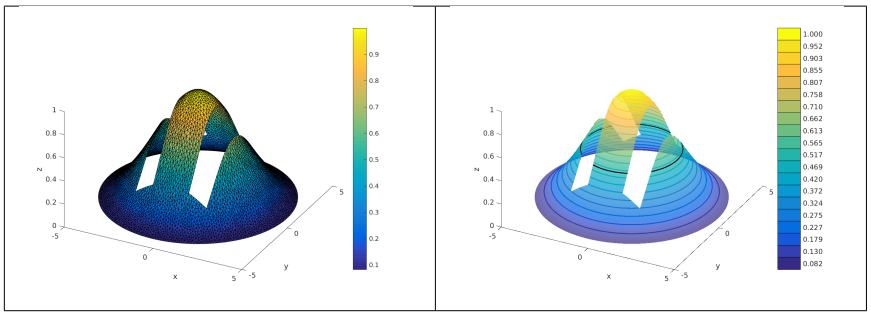
Listing 4: Data representations on a 2D mesh



```
meshfile=gmsh.buildmesh2d('condenser',10);
Th=siMesh(meshfile);
u=@(x,y) cos(2*x-y).*sin(x-2*y);
U=Th.eval(u);
figure(1);
Th.plot(U)
hold on
Th.plotiso(U,'isorange',0, 'color','w', 'Linewidth',1.5);
axis off,axis image,shading interp,colorbar

figure(2)
Th.plot(U,'FaceAlpha',0.8)
hold on
[Colors,isorange]=Th.plotiso(U,'niso',20, 'Linewidth',1.5);
axis off,axis image, shading interp
fc_tools.graphics.colorbarIso(Colors,isorange,'%.3f')
```

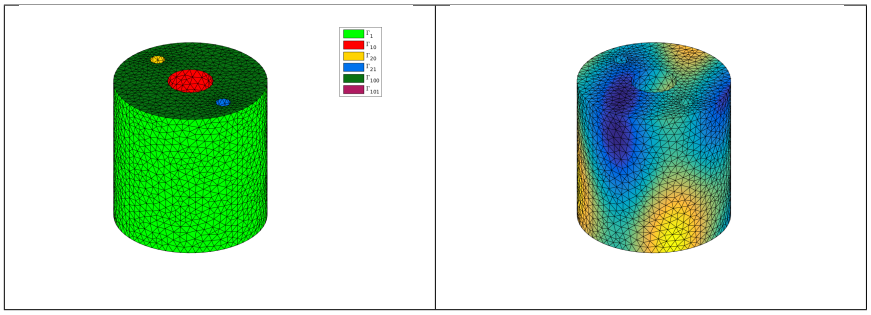
Listing 5: Data representations on a 2D mesh. White lines are zero isolines (left)



```
meshfile=gmsh.buildmesh2d('condenser',7);
Th=siMesh(meshfile);
u=@(x,y)exp(-(x.^2+y.^2)/10);
U=Th.eval(u);
figure(1)
Th.plot(U)
colorbar
view(27,39)
xlabel('x'),ylabel('y'),zlabel('z')
figure(2);
Th.plot(U,'FaceAlpha',0.7)
hold on
[Colors,isorange]=Th.plotiso(U,'niso',20, 'Linewidth',1.5);
Th.plotiso(U,'isorange',0.5, 'color','k', 'Linewidth',1.5);
fc_tools.graphics.colorbarIso(Colors,isorange,'%.3f')
shading interp
view(27,39)
xlabel('x'),ylabel('y'),zlabel('z')
```

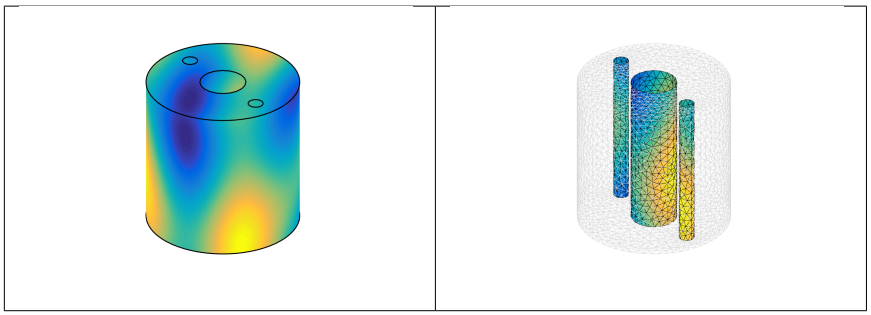
Listing 6: Data representations on a 2D mesh. Black lines are zero isolines (right)

2.2 3D mesh



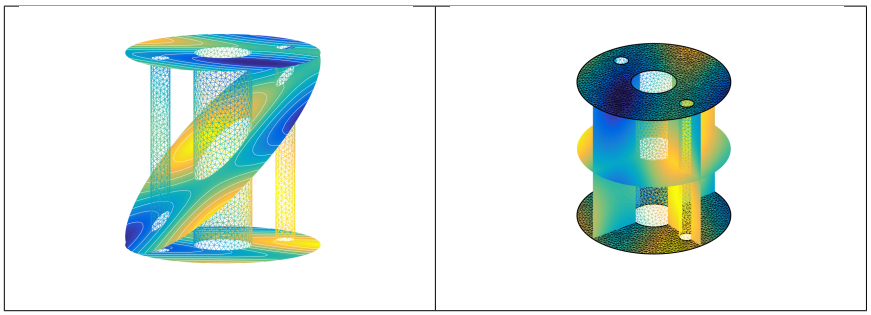
```
meshfile=gmsh.buildmesh3d('cylinder3holes',ns*5);
Th=siMesh(meshfile);
u=@(x,y,z) cos(2*x-y-z).*sin(x-2*y+z);
U=Th.eval(u);
figure(1);
Th.plotmesh('d',2,'legend',true)
axis off;axis image
figure(2)
Th.plot(U,'d',2)
axis off;axis image
```

Listing 7: Data representations on a 3D mesh



```
meshfile=gmsh.buildmesh3d('cylinder3holes',ns*5);
Th=siMesh(meshfile);
u=@(x,y,z) cos(2*x-y-z).*sin(x-2*y+z);
U=Th.eval(u);
figure(1)
Th.plot(U,'d',2)
caxis([min(U),max(U)])
axis off;axis image
shading interp
hold on
Th.plotmesh('d',1, 'color','k', 'Linewidth',1.5)
figure(2)
Th.plotmesh('d',2, 'FaceColor','none', ...
    'EdgeColor',0.85*[1,1,1])
hold on
Th.plot(U,'d',2,'labels',[10,20,21])
caxis([min(U),max(U)])
axis off;axis image
```

Listing 8: Data representations on a 3D mesh



```
meshfile=gmsh.buildmesh3d('cylinder3holes',ns*10);
Th=siMesh(meshfile);
u=@(x,y,z) cos(2*x-y-z).*sin(x-2*y+z);
U=Th.eval(u);
figure(1);
P=fc_tools.graphics.PlaneCoefs([0 0 1],[0 1 1]);
Th.slice(U,P, 'FaceColor','interp', 'EdgeColor','none')
axis off,axis image,hold on
Th.plot(U,'d',2,'labels',[100,101], ...
    'FaceColor','interp', 'EdgeColor','none')
caxis([min(U),max(U)])
isorange=linspace(min(U),max(U),10);
Th.plot(U,'d',2,'labels',[10,20,21], 'FaceColor','none', ...
    'EdgeColor','interp')
Th.plotiso(U,'labels',[100,101], 'isorange',isorange, ...
    'color','w')
Th.sliceiso(U,P,'isorange',isorange, 'color','w')
view(-114,11)
figure(2)
P1=fc_tools.graphics.PlaneCoefs([0 0 1],[1 0 0]);
Th.slice(U,P1, 'FaceColor','interp', 'EdgeColor','none')
hold on
P2=fc_tools.graphics.PlaneCoefs([0 0 1],[0 1 0]);
Th.slice(U,P2, 'FaceColor','interp', 'EdgeColor','none')
P3=fc_tools.graphics.PlaneCoefs([0 0 1],[0 0 1]);
Th.slice(U,P3, 'FaceColor','interp', 'EdgeColor','none')
Th.plot(U,'d',2,'labels',[100,101], 'FaceColor','interp')
Th.plot(U,'d',2,'labels',[10,20,21], 'FaceColor','none', ...
    'EdgeColor','interp')
caxis([min(U),max(U)])
Th.plotmesh('d',1, 'Color','k', 'Linewidth',1.5)
axis off,axis image
```

Listing 9: Data representations on a 3D mesh

2.2.1 Mapping of the unit ball

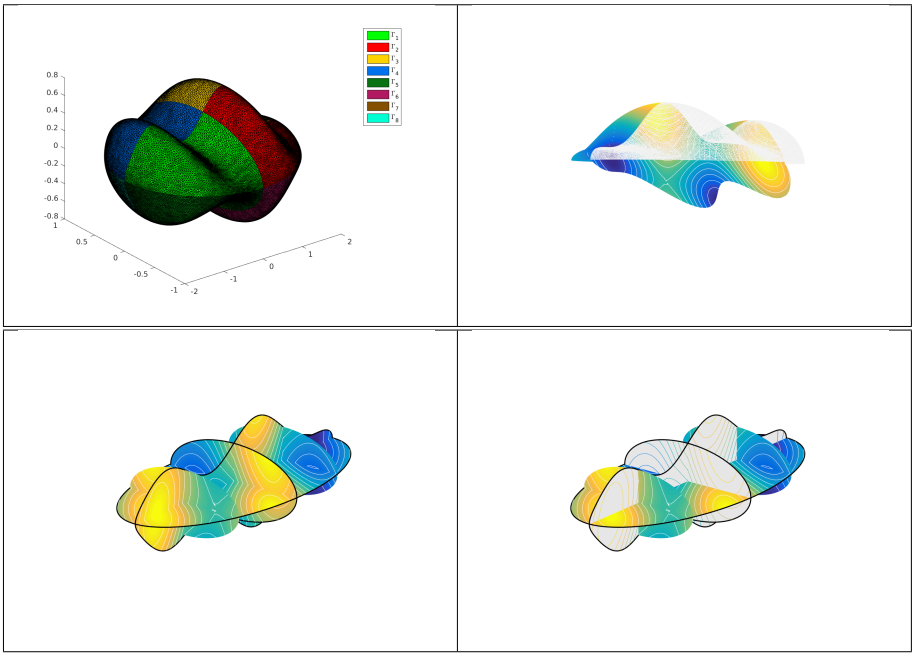
The 3D domain Ω is constructed by mapping a discretization of the unit ball onto Ω by

$$F(y_1, y_2, y_3) = \begin{pmatrix} 2y_1 \\ y_2 \\ \frac{1}{2} y_3 (1 + \frac{1}{2} \sin(2\pi y_1)) \end{pmatrix}, \quad \forall \boldsymbol{y} = (y_1, y_2, y_3), \text{ such that } \|\boldsymbol{y}\|_2 \leqslant 1$$

We represent the function

$$u(x, y, z) = \cos(2x - y - z) \sin(x - 2y + z)$$

on Ω by mapping the unit ball obtained from `gmsh` with `ball8.geo`.



```
trans=@(q)   [2*q(1,:);q(2,:);0.5*q(3,:).*(1+0.5*sin(2*pi*q(1,:)))];
meshfile=gmsh.buildmesh3d('ball8',ns*20);
Th=siMesh(meshfile,'trans',trans);
u=@(x,y,z) cos(2*x-y-z).*sin(x-2*y+z);
U=Th.eval(u);
figure(1)
axis off;axis image
Th.plotmesh('d',2,'legend',true);
figure(2)
P=fc_tools.graphics.PlaneCoefs([0 0 0],[0 1 1]);
Th.slice(U,P)
hold on
Th.plot(U,'d',2,'labels',[1,2], 'EdgeColor','None')
caxis([min(U),max(U)])
isorange=linspace(min(U),max(U),15);
Th.plotmesh('d',2,'labels',[3,4], 'FaceColor','none', ...
    'EdgeColor',0.95*[1,1,1])
Th.plotiso(U,'labels',[3,4],'isorange',isorange, 'color','w' )
Th.sliceiso(U,P,'isorange',isorange, 'color','w')
axis off;axis image
view(127,-1)
figure(3)
Options={'FaceColor','interp', 'EdgeColor','none'};
P1=fc_tools.graphics.PlaneCoefs([0 0 0],[1 0 0]);
Th.slice(U,P1, Options{:} )
hold on
Th.sliceiso(U,P1,'isorange',isorange, 'color','w')
P2=fc_tools.graphics.PlaneCoefs([0 0 0],[0 1 0]);
Th.slice(U,P2,Options{:})
Th.sliceiso(U,P2,'isorange',isorange, 'color','w')
P3=fc_tools.graphics.PlaneCoefs([0 0 0],[0 0 1]);
Th.slice(U,P3,Options{:})
Th.sliceiso(U,P3,'isorange',isorange, 'color','w')
P4=fc_tools.graphics.PlaneCoefs([-1 0 0],[1 0 0]);
Th.slice(U,P4,Options{:})
Th.sliceiso(U,P4,'isorange',isorange, 'color','w')
P5=fc_tools.graphics.PlaneCoefs([1 0 0],[1 0 0]);
Th.slice(U,P5,Options{:})
Th.sliceiso(U,P5,'isorange',isorange, 'color','w')
Th.plotmesh('d',1, 'color','k','Linewidth',1.5)
axis off;axis image
view(-52,20)
figure(4)
P1=fc_tools.graphics.PlaneCoefs([0 0 0],[1 0 0]);
Th.slicemesh(P1,'FaceColor',0.9*[1 1 1], 'EdgeColor','none')
hold on
Th.sliceiso(U,P1,'isorange',isorange)
P2=fc_tools.graphics.PlaneCoefs([0 0 0],[0 1 0]);
Th.slicemesh(P2,'FaceColor',0.9*[1 1 1], 'EdgeColor','none')
Th.sliceiso(U,P2,'isorange',isorange)
P3=fc_tools.graphics.PlaneCoefs([0 0 0],[0 0 1]);
Th.slice(U,P3,'FaceColor','interp')
Th.sliceiso(U,P3,'isorange',isorange, 'color','w')
P4=fc_tools.graphics.PlaneCoefs([-1 0 0],[1 0 0]);
Th.slice(U,P4,'FaceColor','interp')
Th.sliceiso(U,P4,'isorange',isorange,'color','w')
P5=fc_tools.graphics.PlaneCoefs([1 0 0],[1 0 0]);
Th.slice(U,P5,'FaceColor','interp')
Th.sliceiso(U,P5,'isorange',isorange, 'color','w')
Th.plotmesh('d',1,'legend',false, 'color','k', 'Linewidth',1.5)
axis off;axis image;
view(-52,20)
```

Listing 10: Data representations on a 3D mesh

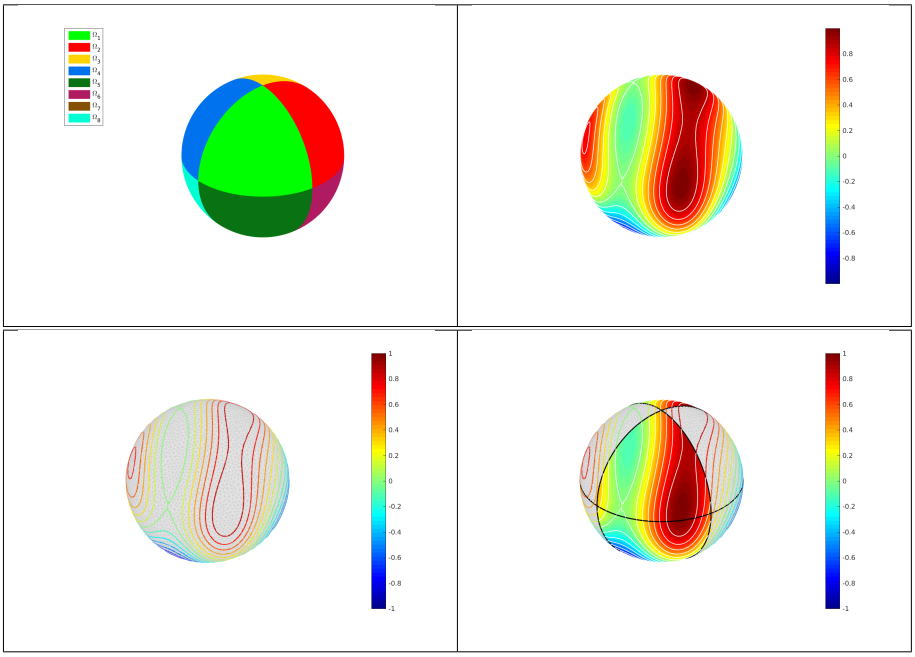
2.3 3D surface meshes

2.3.1 Unit sphere

We represent the function

$$u(x, y, z) = \cos(2x - y - z \sin(x - 2y + z)$$

on the unit sphere obtained from `gmsh` with `sphere8surf.geo`.



```
meshfile=gmsh.buildmesh3ds('sphere8surf',ns*10);
Th=siMesh(meshfile);
u=@(x,y,z) cos(2*x-y-z).*sin(x-2*y+z);
U=Th.eval(u);
figure(1)
Th.plotmesh('legend',true,'EdgeColor','None')
axis off;axis image
set(legend(),'Location','NorthWestOutside')
hfig=figure(2);
Th.plot(U,'d',2)
hold on
Th.plotiso(U,'niso',15,'color',[1 1 1], 'Linewidth',1)
caxis([min(U),max(U)])
axis off, axis image
shading interp,colormap(jet)
h=colorbar;
%set(hfig,'Position',[0 0 800 600])
%set(h,'Position',[0.75 0.25 0.025 0.50])
figure(3)
colormap(jet)
```

```
Th.plotmesh('FaceColor',0.9*[1 1 1], 'EdgeColor',0.8*[1 1 1]);
hold on
Th.plotiso(U,'niso',15, 'Linewidth',1.5)
axis off, axis image
h=colorbar;
%set(hfig,'Position',[0 0 800 600])
%set(h,'Position',[0.75 0.25 0.025 0.50])
hfig=figure(4);
colormap(jet)
Th.plot(U,'d',2,'labels',[1:2:8],'FaceColor','interp','EdgeColor','None')
hold on
Th.plotiso(U,'labels',[1:2:8],'niso',15,'color',[1 1 1], ...
    'Linewidth',1)
Th.plotmesh('labels',[2:2:8], 'FaceColor',0.9*[1 1 1], ...
    'EdgeColor',0.8*[1 1 1]);
Th.plotiso(U,'labels',[2:2:8],'niso',15, 'Linewidth',1.5)
axis off, axis image
Th.plotmesh('d',1,'color',[0 0 0], 'Linewidth',1.5)
h=colorbar;
%set(hfig,'Position',[0 0 800 600])
%set(h,'Position',[0.75 0.25 0.025 0.50])
```

Listing 11: Data representations on a 3D surface mesh

2.3.2 Mapping of the unit sphere

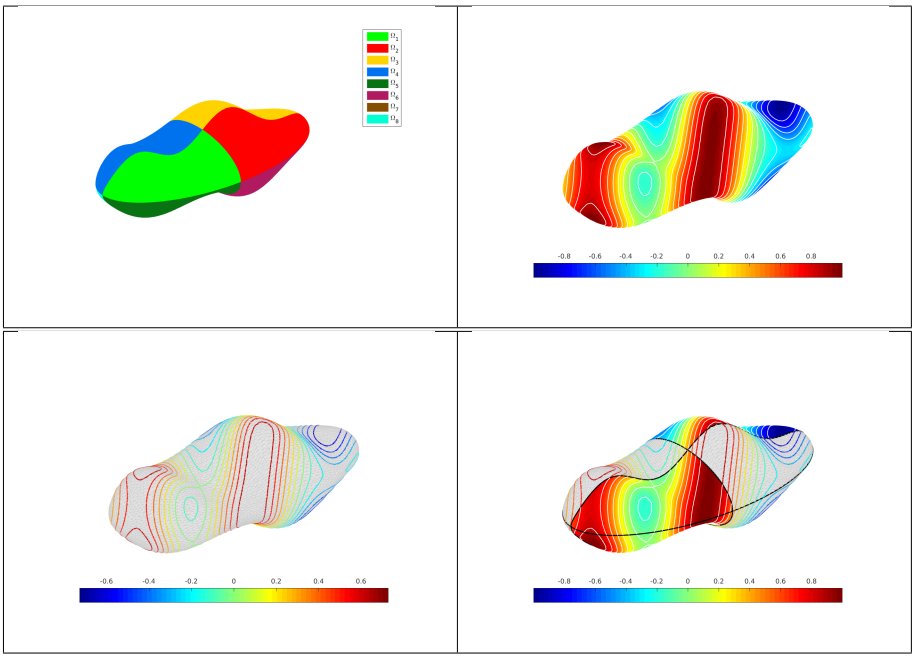
The hypersurface is constructed by mapping a discretization of the unit sphere S^2 onto the surface Ω by

$$F(y_1, y_2, y_3) = \begin{pmatrix} 2y_1 \\ y_2 \\ \frac{1}{2}y_3(1 + \frac{1}{2}\sin(2\pi y_1)) \end{pmatrix}, \quad \forall \boldsymbol{y} = (y_1, y_2, y_3) \in S^2$$

We represent the function

$$u(x, y, z) = \cos(2x - y - z)\sin(x - 2y + z)$$

on the surface Ω by mapping the unit sphere obtained from `gmsh` with `sphere8surf.geo`.

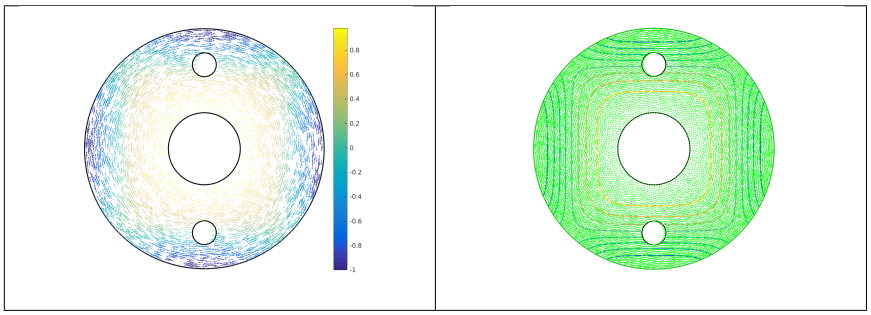

Ω_1
Ω_2
Ω_3
Ω_4
Ω_5
Ω_6
Ω_7
Ω_8
-0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8
-0.6 -0.4 -0.2 0 0.2 0.4 0.6
-0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8

```
meshfile=gmsh.buildmesh3ds('sphere8surf',ns*10);
Th=siMesh(meshfile);
u=@(x,y,z) cos(2*x-y-z).*sin(x-2*y+z);
U=Th.eval(u);
figure(1)
Th.plotmesh('legend',true,'EdgeColor','None')
axis off;axis image
set(legend(),'Location','NorthWestOutside')
hfig=figure(2);
Th.plot(U,'d',2)
hold on
Th.plotiso(U,'niso',15,'color',[1 1 1], 'Linewidth',1)
caxis([min(U),max(U)])
axis off, axis image
shading interp,colormap(jet)
h=colorbar;
%set(hfig,'Position',[0 0 800 600])
%set(h,'Position',[0.75 0.25 0.025 0.50])
figure(3)
colormap(jet)
Th.plotmesh('FaceColor',0.9*[1 1 1], 'EdgeColor',0.8*[1 1 1]);
hold on
Th.plotiso(U,'niso',15, 'Linewidth',1.5)
axis off, axis image
h=colorbar;
%set(hfig,'Position',[0 0 800 600])
%set(h,'Position',[0.75 0.25 0.025 0.50])
hfig=figure(4);
colormap(jet)
Th.plot(U,'d',2,'labels',[1:2:8],'FaceColor','interp','EdgeColor','None')
hold on
Th.plotiso(U,'labels',[1:2:8],'niso',15,'color',[1 1 1], 'Linewidth',1)
Th.plotmesh('labels',[2:2:8], 'FaceColor',0.9*[1 1 1], ...
    'EdgeColor',0.8*[1 1 1]);
Th.plotiso(U,'labels',[2:2:8],'niso',15, 'Linewidth',1.5)
axis off, axis image
Th.plotmesh('d',1,'color',[0 0 0], 'Linewidth',1.5)
h=colorbar;
%set(hfig,'Position',[0 0 800 600])
%set(h,'Position',[0.75 0.25 0.025 0.50])
```

Listing 12: Data representations on a 3D surface mesh

2.4 Vector field representation on meshes

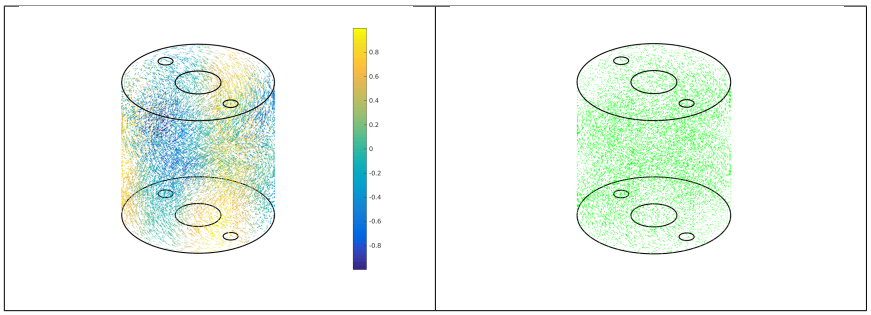
2.4.1 2D mesh



```
meshfile=gmsh.buildmesh2d('disk3holes',50);
Th=siMesh(meshfile);
u=@(x,y) cos(pi*x.^2).*cos(pi*y.^2);
U=Th.eval(u);
w={@(x,y) y.*cos(-(x.^2+y.^2)/10);@(x,y) ...
    -x.*cos(-(x.^2+y.^2)/10)};
W=[Th.eval(w{1}),Th.eval(w{2})]';
figure(1);
Th.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on
Th.plotVectorField(W,'colordata',U,'freq',2, 'scale',0.05)
axis off;axis image
colorbar
figure(2)
Th.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on
Th.plotVectorField(W,'LineWidth',1)
Th.plotiso(U, 'LineWidth',1.5)
axis off;axis image
colormap('jet')
```

Listing 13: Vector field representations on a 2D mesh

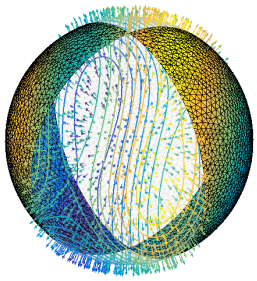
2.4.2 3D mesh



```
meshfile=gmsh.buildmesh3d('cylinder3holes',ns*10);
Th=siMesh(meshfile);
u=@(x,y,z) cos(2*x-y-z).*sin(x-2*y+z);
U=Th.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) sin(2*pi*z)};
W=[Th.eval(w{1}),Th.eval(w{2}),Th.eval(w{3})];
figure(1);
Th.plotmesh('d',1, 'color','k', 'Linewidth',1.5)
hold on
Th.plotVectorField(W,'colordata',U, 'freq',3,'scale',0.05)
view(3)
axis off;axis image
colorbar
figure(2)
Th.plotmesh('d',1, 'color','k', 'Linewidth',1.5)
hold on
Th.plotVectorField(W,'freq',3)
view(3)
axis off;axis image
```

Listing 14: Vector field representations on a 3D mesh

2.4.3 3D surface mesh



```
meshfile=gmsh.buildmesh3ds('sphere8surf',ns*10);
Th=siMesh(meshfile);
u=@(x,y,z) cos(2*x-y-z).*sin(x-2*y+z);
U=Th.eval(u);
w={@(x,y,z) x;@(x,y,z) y; @(x,y,z) z};
W=[Th.eval(w{1}),Th.eval(w{2}),Th.eval(w{3})];
figure(1)
Th.plotmesh('labels',1:2:8, 'FaceColor','none', ...
    'EdgeColor',0.8*[1 1 1], 'EdgeAlpha',0.3);
hold on
Th.plotiso(U,'labels',[1:2:8], 'niso',15, 'Linewidth',1);
Th.plotVectorField(W,'colordata',U, 'labels',1:2:8, ...
    'freq',3,'scale',0.1)
Th.plot(U,'d',2,'labels',[2:2:8]);
axis off;axis image
```

Listing 15: Vector field representation on a 3D surface mesh

3 Functions of the fc-simesh toolbox

3.1 siMesh methods

3.1.1 siMesh constructor

The constructor of the siMesh class can initialize the object from various kind of mesh file format : *.msh* (default `gmsh` format), *.mesh* (`FreeFEM++` or `Medit`) or ... (`triangle`).

Syntaxe

```
Th=siMesh(meshfile)

Th=siMesh(meshfile,Name,Value)
```

Description

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

Th=siMesh(meshfile,Name,Value, ...) specifies function options using one or more Name,Value pair arguments. The Name options can be

- `'format'` : to specify the format of the mesh file meshfile. Value must be *'medit'*, *'gmsh'* (default), *'freefem'* or *'triangle'*.
- `'dim'` : to specify the space dimension (default 2),
- `'d'` : to specify the dimensions of the simplices to read, (default [dim,dim−1])

Examples The following example use the function gmsh.buildmesh2d of the `fc-oogmsh` toolboxto build the mesh from the *.geo* file `condenser11.geo`. This .geo file is located in the directory `geodir` of the `fc-oogmsh` toolbox.

Matlab commands with output

```
meshfile=gmsh.buildmesh2d('condenser11',25,'verbose',0);
disp('*** Read mesh ***')
Th=siMesh(meshfile)
```

```
*** Read mesh ***

Th =

  siMesh with properties:
            d: 2 double
          dim: 2 double
           sTh: (1x19 cell)
          nsTh: 19 double
     toGlobal: (1x2990 double)
     toParent: (1x2990 double)
      sThsimp: (1x19 double)
       sThlab: (1x19 double)
    sThcolors: (19x3 double)
         bbox: [ -2 2 -2 2 ] (1x4 double)
    sThgeolab: []
   sThphyslab: (1x6 double)
  sThpartlabs: []
           nq: 2990 double
```

3.1.2 function plotmesh

The method plotmesh displays the mesh or parts of the mesh defined by an siMesh object.

Syntaxe

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

Description

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

`Th.plotmesh(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),
- `'legend'` : add a legend to graph if true (default : false)
- `'bounds'` : If true, draw the borders of the selected elementaries mesh elements (only for 2-dimensional simplices). (default : false)
- `'cutPlan'` : cut mesh by n plans given by n-by-4 array P where the equation of the i-th cut plan is given by

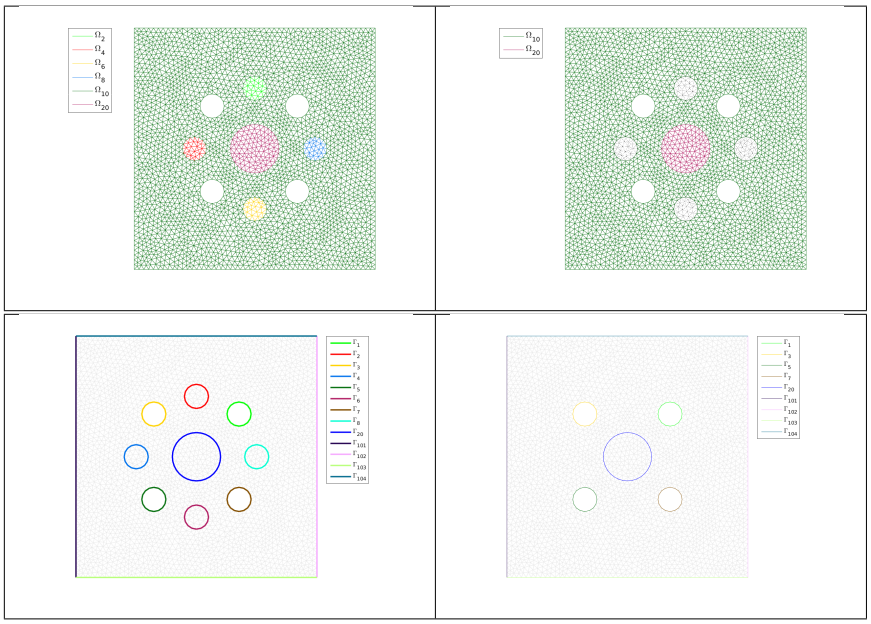
$$P(i,1)\,x + P(i,2)\,y + P(i,3)\,z + P(i,4) = 0.$$

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

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

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

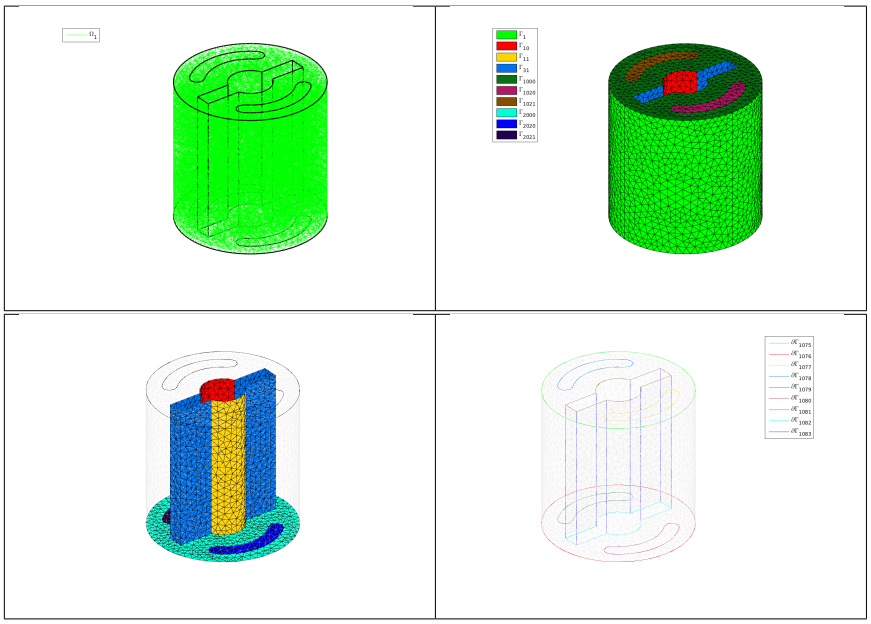
2D example The following example use the *.geo* file `condenser11.geo` which is in the directory `geodir` of the toolbox ....



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

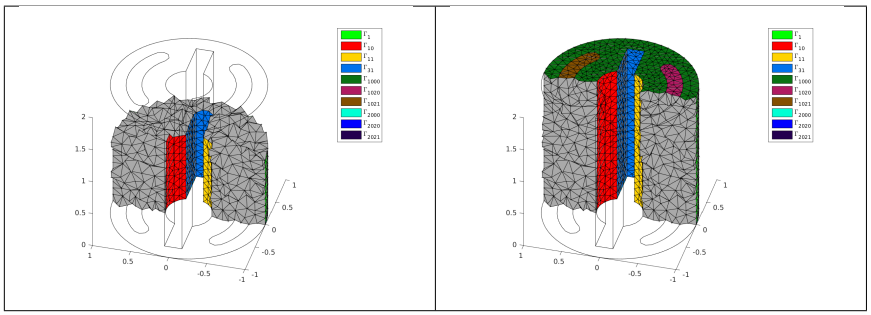
Listing 16: 2D plot mesh

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



```
meshfile=gmsh.buildmesh3d('cylinderkey03',ns*5,'verbose',0, ...
    'geodir',geodir3d);
Th=siMesh(meshfile);
figure(1)
Th.plotmesh('legend',true)
hold on
Th.plotmesh('d',1,'Color','k','Linewidth',1.5)
axis off;axis image
set(legend(),'Location','NorthWestOutside')
figure(2)
Th.plotmesh('d',2,'legend',true)
view(3);hold on;axis off;axis image
set(legend(),'Location','NorthWestOutside')
figure(3)
Th.plotmesh('d',2,'labels',[1,1000,1020,1021], ...
    'EdgeColor',0.9*[1,1,1], ...
    'EdgeAlpha',0.4,'FaceColor','none')
hold on;axis off;axis image
Th.plotmesh('d',2,'labels',1000,'bounds',true,'color','k')
Th.plotmesh('d',2,'labels',[10,11,31,2000,2020,2021])
figure(4)
Th.plotmesh('d',2, 'EdgeColor',0.9*[1,1,1], ...
    'EdgeAlpha',0.4,'FaceColor','none')
hold on;axis off;axis image
Th.plotmesh('d',1,'legend',true)
```

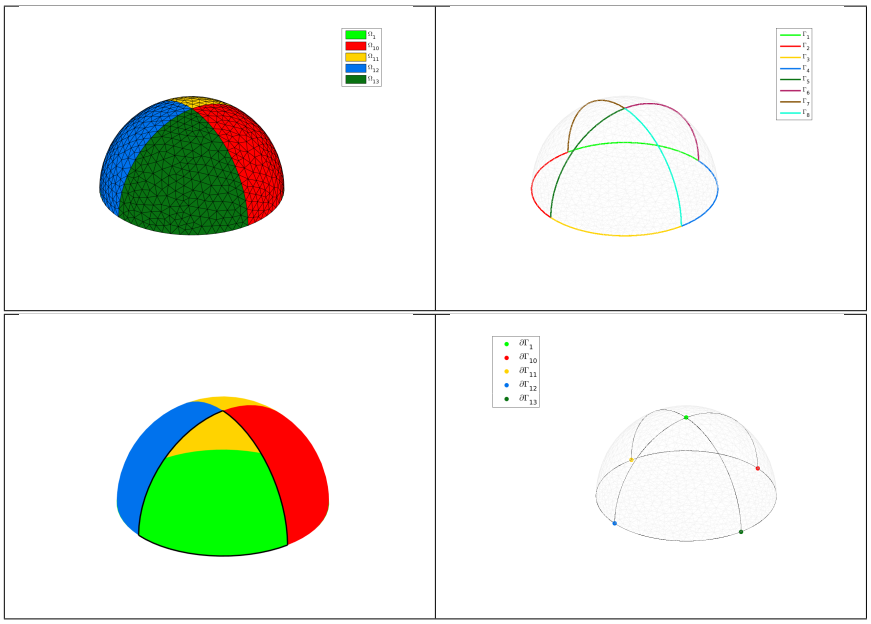
Listing 17: 3D plot mesh



```
meshfile=gmsh.buildmesh3d('cylinderkey03',ns*5,'verbose',0, ...
    'geodir',geodir3d);
Th=siMesh(meshfile);
figure(5)
P=[fc_tools.graphics.PlaneCoefs([0 0 1], [0 0 1]); ...
   fc_tools.graphics.PlaneCoefs([0 0 1],[-1 0 0])];
Th.plotmesh('d',1,'color','black')
hold on;axis image
Th.plotmesh('cutPlan',P,'Color','DarkGrey')
Th.plotmesh('d',2,'cutPlan',P,'legend',true);
view(-75,36)
figure(6)
P=fc_tools.graphics.PlaneCoefs([0 0 1],[-1 0 0]);
Th.plotmesh('d',1,'color','black')
hold on;axis image
Th.plotmesh('cutPlan',P,'Color','DarkGrey')
Th.plotmesh('d',2,'cutPlan',P,'legend',true);
view(-75,36)
```

Listing 18: 3D plot mesh,

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



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

Listing 19: 3D surface mesh : plot function

3.1.3 function PLOT

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

Syntaxe

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

Description

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

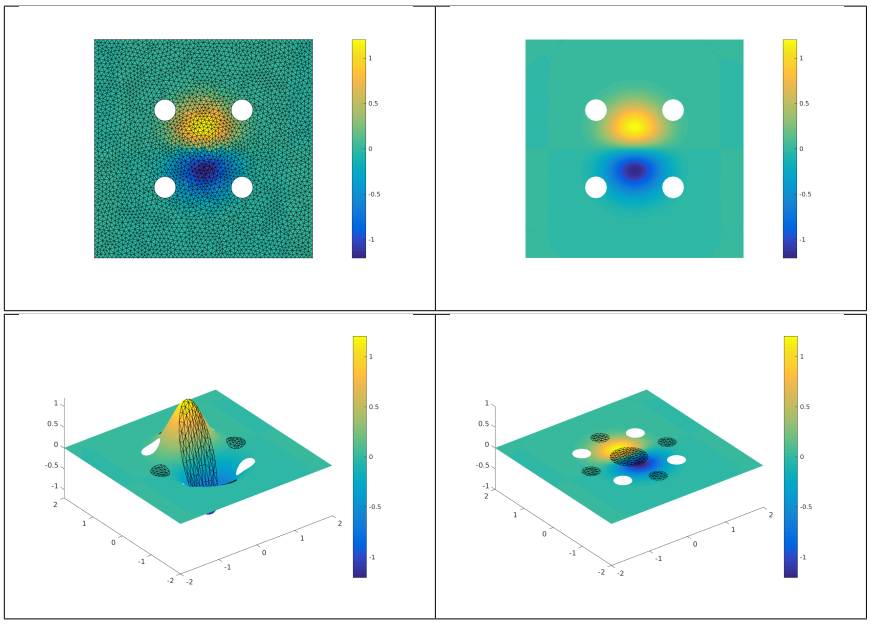
Th.**plot**(u,Name,Value, ...) specifies function options using one or more Name,Value pair arguments. Options of first level are

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

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

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

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



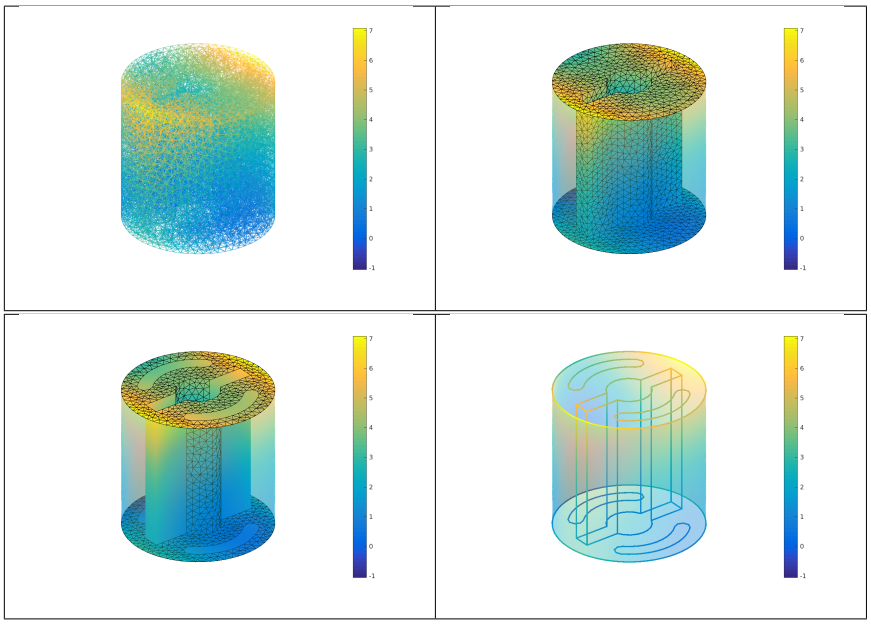
```
meshfile=gmsh.buildmesh2d('condenser11',25,'verbose',0);
Th=siMesh(meshfile);
u=Th.eval(@(x,y) 5*exp(-3*(x.^2+y.^2)).*cos(x).*sin(y));
figure(1)
Th.plot(u)
axis off;axis equal;
colorbar
figure(2)
Th.plot(u)
axis off;axis equal;
shading interp;colorbar

figure(3)
Th.plot(u,'labels',[2:2:8,20],'FaceColor','interp')
view(3);hold on;
Th.plot(u,'labels',10,'FaceColor','interp','EdgeColor','none')
axis equal;colorbar

figure(4)
Th.plot(u,'labels',[2:2:8,20], 'plan',true ...
    ,'FaceColor','interp')
view(3);hold on;
Th.plot(u,'labels',10, 'plan',true, ...
    'FaceColor','interp','EdgeColor','none')
axis equal;colorbar
```

Listing 20: 2D mesh : plotVal function

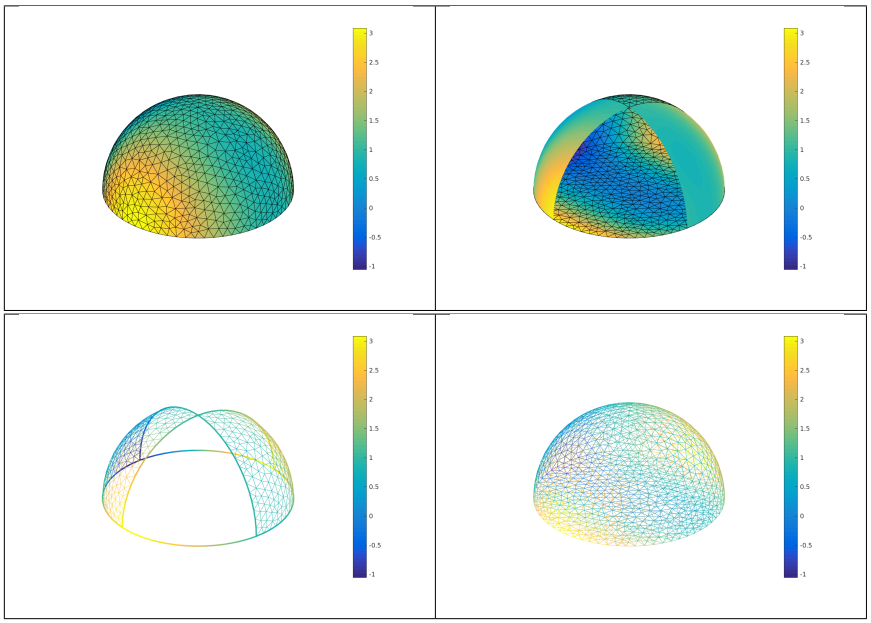
3D example The following example use the *.geo* file `cylinderkey.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
meshfile=gmsh.buildmesh3d('cylinderkey03',ns*5, ...
    'geodir',geodir3d);
Th=siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
Th.plot(u);
axis off;axis equal;colorbar
figure(2)
Th.plot(u,'d',2,'labels',[10,11,31,1000,1020,1021,2000,2020,2021])
hold on
Th.plot(u,'d',2,'labels',1,'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
axis off;axis equal;colorbar
figure(3)
Th.plot(u,'d',2,'labels',1,'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
hold on
Th.plot(u,'d',2,'labels',[10,11,1000,2000])
Th.plot(u,'d',2,'labels',[31,1020,1021,2020,2021], ...
    'FaceColor','interp','EdgeColor','none')
axis off;axis equal;colorbar
figure(4)
Th.plot(u,'d',2,'labels',1,'FaceColor','interp', ...
    'EdgeColor','none','FaceAlpha',0.4)
hold on
Th.plot(u,'d',1,'LineWidth',2)
axis off;axis equal;colorbar
```

Listing 21: 3D mesh : plotVal function

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



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

Listing 22: 3D surface mesh : plotVal function

3.1.4 function PLOTISO

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

in space dimension 2 or 3.

Syntaxe

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

Description

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

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

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

The options of second level are all options of

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

This function accepts until 4 output arguments :

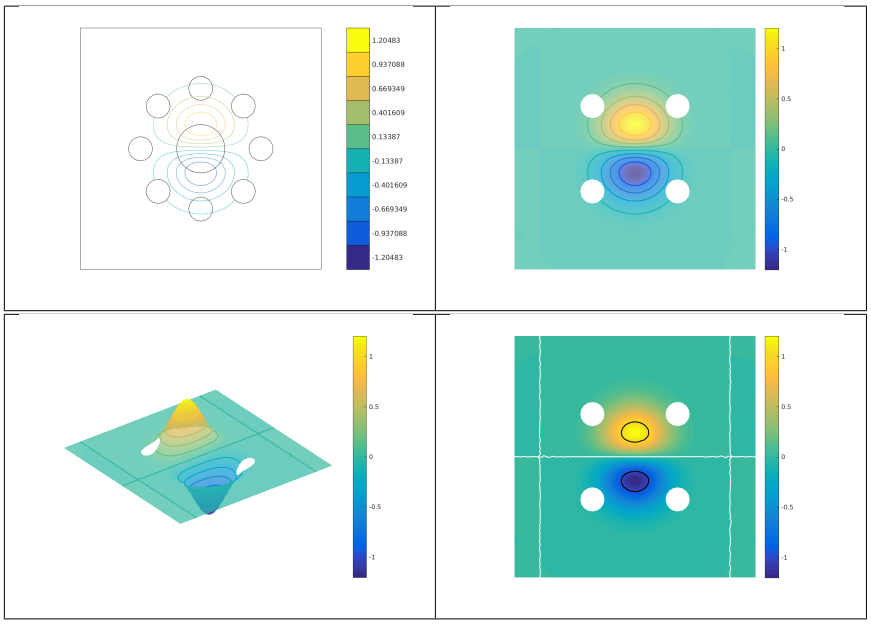
bullet 1st output is the colors of the isolines

bullet 2nd output is the isovalues of the isolines

bullet 3th output is the handle of the colobar iso.

bullet 4th output is all the handles of the isolines as an 2D-array of dimension N-by-niso, where N is the number of 2-simplex elementary meshes where isolines are drawn.

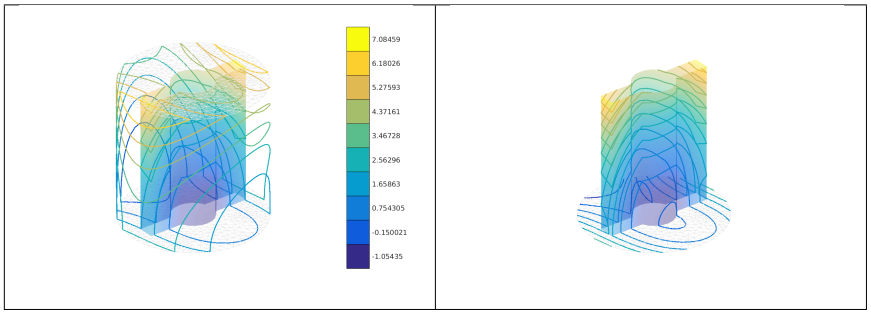
2D example The following example use the *.geo* file `condenser11.geo` which is in the directory `geodir` of the toolbox.



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

Listing 23: 2D mesh : plotIsolines function

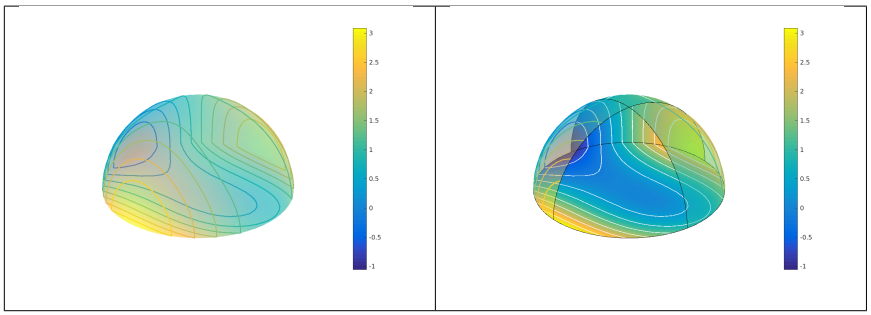
3D example The following example use the *.geo* file `cylinderkey.geo` which is in the directory `geodir` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.




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

Listing 24: 3D mesh : plotIsolines function

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



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

Listing 25: 3D surface mesh : plotIsolines function

3.1.5 function SLICEMESH

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

Syntaxe

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

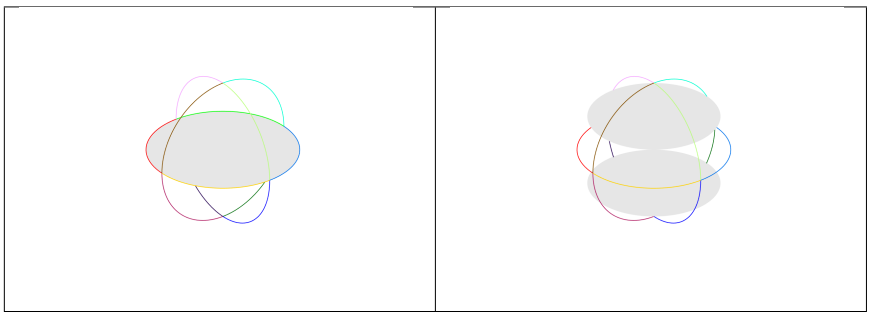
Description

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

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

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

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



```
meshfile=gmsh.buildmesh3d('ball8',ns*10,'verbose',0);
Th=siMesh(meshfile);
figure(1)
Th.plotmesh('d',1,'LineWidth',1)
hold on
P=fc_tools.graphics.PlaneCoefs([0 0 0],[0 0 1]);
Th.slicemesh(P,'FaceColor',0.9*[1 1 1])
axis off;axis image;
figure(2)
Th.plotmesh('d',1,'LineWidth',1);
hold on
P2=fc_tools.graphics.PlaneCoefs([0 0 -0.5],[0 0 1]);
Th.slicemesh(P2,'FaceColor',0.9*[1 1 1], ...
    'EdgeColor','none')
P3=fc_tools.graphics.PlaneCoefs([0 0 0.5],[0 0 1]);
Th.slicemesh(P3,'FaceColor',0.9*[1 1 1], ...
    'EdgeColor','none')
axis off;axis image;
```

Listing 26: 3D mesh : slicemesh function

3.1.6 function SLICE

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

Syntaxe

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

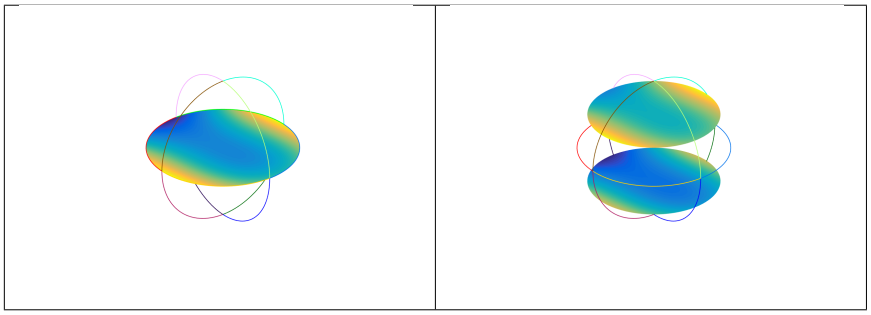
Description

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

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

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

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



```
meshfile=gmsh.buildmesh3d('ball8',ns*10,'verbose',0);
Th=siMesh(meshfile);
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y+z);
figure(1)
Th.plotmesh('d',1,'LineWidth',1)
hold on
P=fc_tools.graphics.PlaneCoefs([0 0 0],[0 0 1]);
Th.slice(u,P,'Facecolor','interp')
axis off;axis image;
figure(2)
Th.plotmesh('d',1,'LineWidth',1);
hold on
P2=fc_tools.graphics.PlaneCoefs([0 0 -0.5],[0 0 1]);
Th.slice(u,P2,'Facecolor','interp')
P3=fc_tools.graphics.PlaneCoefs([0 0 0.5],[0 0 1]);
Th.slice(u,P3,'Facecolor','interp')
axis off;axis image;
```

Listing 27: 3D mesh : **slice** function

3.1.7 function SLICEISO

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

Syntaxe

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

Description

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

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

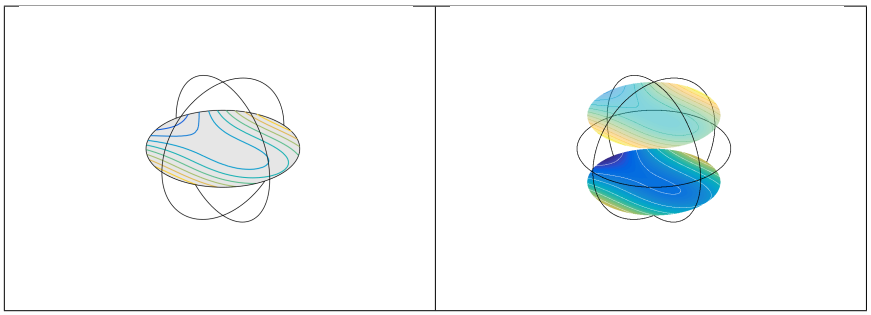
- `'labels'` : to select the labels of the elements to intersect,
- `'niso'` : to specify the number of isolines (default : 10)
- `'isorange'` : to specify the list of isovalues (default : empty)
- `'color'` : to specify one color for all isolines (default : empty)
- `'isocolorbar'` : if true display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

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

This function accepts until 4 output arguments :

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

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



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

Listing 28: 3D mesh : isoslice function

3.1.8 function plotVectorField

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

Syntaxe

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

Description

`Th.plotVectorField(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.

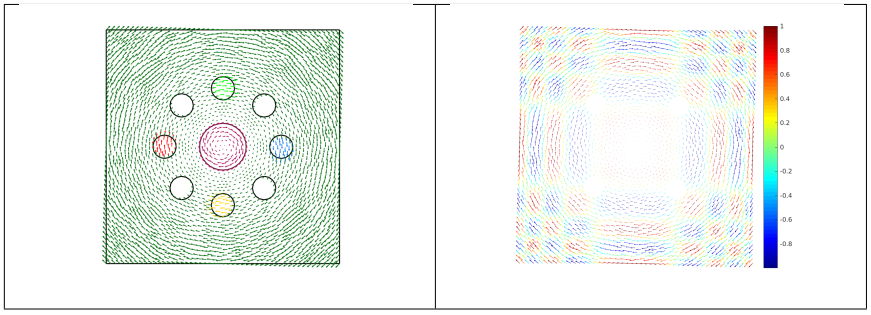
`Th.plotVectorField(V,Name,Value, ...)` specifies function options using one or more Name,Value pair arguments. Options of first level are

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

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

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

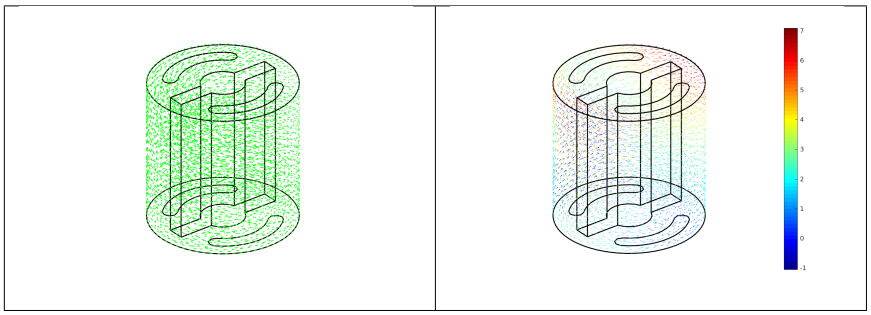
2D example The following example use the *.geo* file `condenser11.geo` which is in the directory `geodir` of the toolbox.



```
meshfile=gmsh.buildmesh2d('condenser11',25,'verbose',0);
Th=siMesh(meshfile);
u=@(x,y) cos(pi*x.^2).*cos(pi*y.^2);
U=Th.eval(u);
w={@(x,y) y.*cos(-(x.^2+y.^2)/10);@(x,y) ...
    -x.*cos(-(x.^2+y.^2)/10)};
W=[Th.eval(w{1}),Th.eval(w{2})]';
figure(1)
Th.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on
Th.plotVectorField(W,'LineWidth',1)
axis off;axis image
figure(2)
Th.plotmesh('d',1,'color','k', 'LineWidth',1.5)
Th.plotVectorField(W,'colordata',U, 'scale',0.05)
axis off;axis image
colormap('jet');colorbar
```

Listing 29: 2D mesh : plotVectorField function

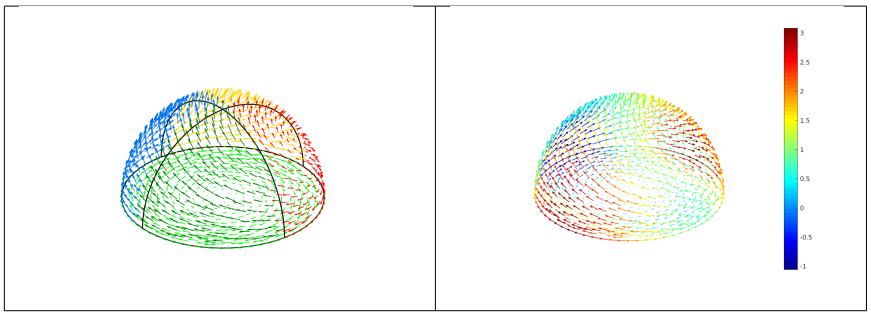
3D example The following example use the *.geo* file `cylinderkey03.geo` which is in the directory `geodir/3d` of the toolbox. This file contains description of a 3D mesh with simplices of dimensions 1, 2 and 3.



```
meshfile=gmsh.buildmesh3d('cylinderkey03',ns*5, ...
    'geodir',geodir3d);
Th=siMesh(meshfile);
w={@(x,y,z) y.*cos(-(x.^2+y.^2)/10);@(x,y,z) ...
    -x.*cos(-(x.^2+y.^2)/10);@(x,y,z) z/5 };
W=[Th.eval(w{1}),Th.eval(w{2}),Th.eval(w{3})]';
u=Th.eval(@(x,y,z) 3*x.^2-y.^3+z.^2+x.*y);
figure(1)
Th.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on
Th.plotVectorField(W,'LineWidth',1)
axis off;axis image
figure(2)
Th.plotmesh('d',1,'color','k', 'LineWidth',1.5)
hold on
Th.plotVectorField(W,'colordata',u, 'scale',0.05)
axis off;axis image
colormap('jet');colorbar
```

Listing 30: 3D mesh : plotVectorField function

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



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

Listing 31: 3D surface mesh : plotVectorField function