FC-VFEM$\mathbb{P}_1$ Octave package, User's Guide [1]

François Cuvelier[2]

2017/10/15

[1]Compiled with Octave 4.2.1

[2]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 supported by the ANR project DEDALES under grant ANR-14-CE23-0005.

Abstract

FC-VFEM$\mathbb{P}_1$ is an object-oriented Octave package dedicated to solve scalar or vector boundary value problem (BVP) by $\mathbb{P}_1$-Lagrange finite element methods in any space dimension. It integrates the FC-SIMESH package which allows a great flexibility in graphical representations of the meshes and datas on the meshes.

This package also contains the techniques of vectorization presented in [2] and extended in [1] and allows good performances when using finite elements methods.

Contents

Chapter 1

Generic Boundary Value Problems

The notations of [4] are employed in this section and extended to the vector case.

1.1 Scalar boundary value problem

Let Ω be a bounded open subset of $\mathbb{R}^d$, $d \geqslant 1$. The boundary of Ω is denoted by Γ.

We denote by $\mathcal{L}_{\mathbb{A},\boldsymbol{b},\boldsymbol{c},a_0} = \mathcal{L} : \mathrm{H}^2(\Omega) \longrightarrow L^2(\Omega)$ the second order linear differential operator acting on *scalar fields* defined, $\forall u \in \mathrm{H}^2(\Omega)$, by

$$\mathcal{L}_{\mathbb{A},\boldsymbol{b},\boldsymbol{c},a_0}(u) \stackrel{\mathsf{def}}{=} -\operatorname{div}(\mathbb{A}\boldsymbol{\nabla} u) + \operatorname{div}(\boldsymbol{b}u) + \langle \boldsymbol{\nabla} u, \boldsymbol{c}\rangle + a_0 u \tag{1.1}$$

where $\mathbb{A} \in (L^\infty(\Omega))^{d\times d}$, $\boldsymbol{b} \in (L^\infty(\Omega))^d$, $\boldsymbol{c} \in (L^\infty(\Omega))^d$ and $a_0 \in L^\infty(\Omega)$ are given functions and $\langle\cdot,\cdot\rangle$ is the usual scalar product in $\mathbb{R}^d$. We use the same notations as in the chapter 6 of [4] and we note that we can omit either $\operatorname{div}(\boldsymbol{b}u)$ or $\langle \boldsymbol{\nabla} u, \boldsymbol{c}\rangle$ if $\boldsymbol{b}$ and $\boldsymbol{c}$ are sufficiently regular functions. We keep both terms with $\boldsymbol{b}$ and $\boldsymbol{c}$ to deal with more boundary conditions. It should be also noted that it is important to preserve the two terms $\boldsymbol{b}$ and $\boldsymbol{c}$ in the generic formulation to enable a greater flexibility in the choice of the boundary conditions.

Let Γ^D, Γ^R be open subsets of Γ, possibly empty and $f \in L^2(\Omega)$, $g^D \in \mathrm{H}^{1/2}(\Gamma^D)$, $g^R \in L^2(\Gamma^R)$, $a^R \in L^\infty(\Gamma^R)$ be given data.

A *scalar* boundary value problem is given by

Scalar BVP 1 : generic problem

Find $u \in \mathrm{H}^2(\Omega)$ such that

$$\mathcal{L}(u) = f \quad \text{in } \Omega, \tag{1.2}$$

$$u = g^D \quad \text{on } \Gamma^D, \tag{1.3}$$

$$\frac{\partial u}{\partial n_{\mathcal{L}}} + a^R u = g^R \quad \text{on } \Gamma^R. \tag{1.4}$$

The **conormal derivative** of u is defined by

$$\frac{\partial u}{\partial n_{\mathcal{L}}} \stackrel{\mathsf{def}}{=} \langle \mathbb{A}\boldsymbol{\nabla} u, \boldsymbol{n}\rangle - \langle \boldsymbol{b}u, \boldsymbol{n}\rangle \tag{1.5}$$

The boundary conditions (1.3) and (1.4) are respectively **Dirichlet** and **Robin** boundary conditions. **Neumann** boundary conditions are particular Robin boundary conditions with $a^R \equiv 0$.

To have an outline of the FC-VFEM$\mathbb{P}_1$ package, a first and simple problem is quickly present. Explanations will be given in next sections.

The problem to solve is the Laplace problem for a condenser.

Usual BVP 1 : 2D condenser problem

Find $u \in \mathrm{H}^2(\Omega)$ such that

$$
\begin{aligned}
-\Delta u &= 0 \quad \text{in } \Omega \subset \mathbb{R}^2, &(1.6)\\
u &= 0 \quad \text{on } \Gamma_1, &(1.7)\\
u &= -12 \quad \text{on } \Gamma_{98}, &(1.8)\\
u &= 12 \quad \text{on } \Gamma_{99}, &(1.9)
\end{aligned}
$$

where Ω and its boundaries are given in Figure 1.1.

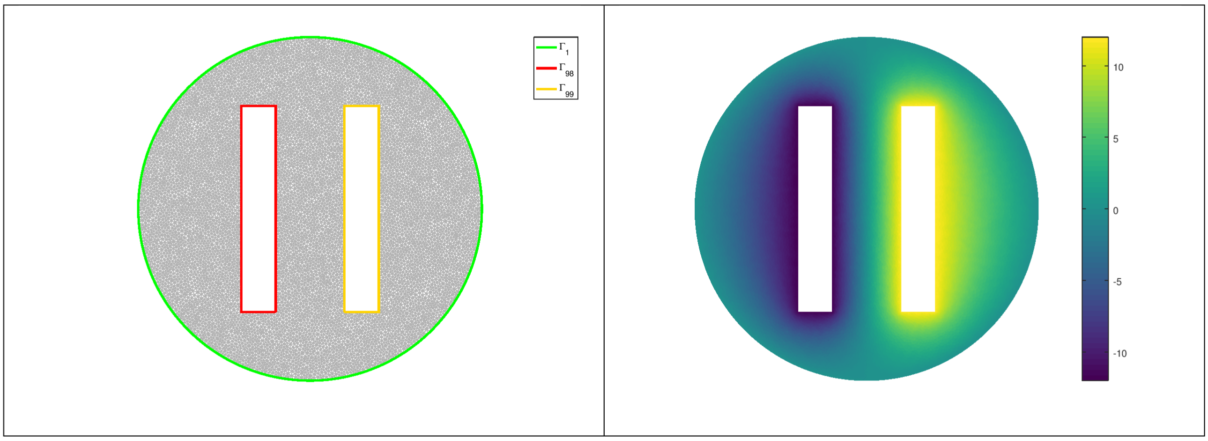


Figure 1.1: 2D condenser mesh and boundaries (left) and numerical solution (right)

The problem (1.6)-(1.9) can be equivalently expressed as the scalar BVP (1.2)-(1.4) :

Scalar BVP 2 : 2D condenser problem

Find $u \in \mathrm{H}^2(\Omega)$ such that

$$
\begin{aligned}
\mathcal{L}(u) &= f && \text{in } \Omega,\\
u &= g^D && \text{on } \Gamma^D = \Gamma_1 \cup \Gamma_{98} \cup \Gamma_{99}.
\end{aligned}
$$

where $\mathcal{L} := \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}$, $f \equiv 0$, and

$$g^D := 0 \text{ on } \Gamma_1,\ g^D := -12 \text{ on } \Gamma_{98},\ g^D := +12 \text{ on } \Gamma_{99}$$

In Listing 19 a complete code is given to solve this problem.

```
meshfile=gmsh.buildmesh2d('condenser',10); % generate mesh
Th=siMesh(meshfile); % read mesh
Lop=Loperator(2,2,{1,0;0,1},[],[],[]);
pde=PDEelt(Lop);
bvp=BVP(Th,pde);
bvp.setDirichlet( 1, 0.);
bvp.setDirichlet( 98, -12.);
bvp.setDirichlet( 99, +12.);
U=bvp.solve();
% Graphic parts
figure(1)
Th.plotmesh('color',0.7*[1,1,1])
hold on
Th.plotmesh('d',1,'Linewidth',2,'legend',true)
axis off,axis image
figure(2)
```

```
Th.plot(U,'edgecolor','none','facecolor','interp')
axis off,axis image;colorbar
```

Listing 1.1: Complete Octave code to solve the 2D condenser problem with graphical representations

Obviously, more complex problems will be studied in section **??** and complete explanations on the code will be given in next sections. Previously, the vector BVP is formally presented with an application.

1.2 Vector boundary value problem

Let $m \geqslant 1$ and $\mathcal{H}$ be the m-by-m matrix of second order linear differential operators defined by

$$\left\{ \begin{array}{lllll} \mathcal{H} & : & \left(\mathrm{H}^2(\Omega)\right)^m & \longrightarrow & \left(L^2(\Omega)\right)^m \\ & & \boldsymbol{u} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m) & \longmapsto & \boldsymbol{f} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_m) \stackrel{\text{def}}{=} \mathcal{H}(\boldsymbol{u}) \end{array} \right. \tag{1.10}$$

where

$$\boldsymbol{f}_\alpha = \sum_{\beta=1}^{m} \mathcal{H}_{\alpha,\beta}(\boldsymbol{u}_\beta), \quad \forall \alpha \in [\![1, m]\!], \tag{1.11}$$

with, for all $(\alpha, \beta) \in [\![1, m]\!]^2$,

$$\mathcal{H}_{\alpha,\beta} \stackrel{\text{def}}{=} \mathcal{L}_{\mathbb{A}^{\alpha,\beta}, \boldsymbol{b}^{\alpha,\beta}, \boldsymbol{c}^{\alpha,\beta}, a_0^{\alpha,\beta}} \tag{1.12}$$

and $\mathbb{A}^{\alpha,\beta} \in (L^\infty(\Omega))^{d\times d}$, $\boldsymbol{b}^{\alpha,\beta} \in (L^\infty(\Omega))^d$, $\boldsymbol{c}^{\alpha,\beta} \in (L^\infty(\Omega))^d$ and $a_0^{\alpha,\beta} \in L^\infty(\Omega)$ are given functions. We can also write in matrix form

$$\mathcal{H}(\boldsymbol{u}) = \begin{pmatrix} \mathcal{L}_{\mathbb{A}^{1,1}, \boldsymbol{b}^{1,1}, \boldsymbol{c}^{1,1}, a_0^{1,1}} & \cdots & \mathcal{L}_{\mathbb{A}^{1,m}, \boldsymbol{b}^{1,m}, \boldsymbol{c}^{1,m}, a_0^{1,m}} \\ \vdots & \ddots & \vdots \\ \mathcal{L}_{\mathbb{A}^{m,1}, \boldsymbol{b}^{m,1}, \boldsymbol{c}^{m,1}, a_0^{m,1}} & \cdots & \mathcal{L}_{\mathbb{A}^{m,m}, \boldsymbol{b}^{m,m}, \boldsymbol{c}^{m,m}, a_0^{m,m}} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_m \end{pmatrix}. \tag{1.13}$$

We remark that the $\mathcal{H}$ operator for $m = 1$ is equivalent to the $\mathcal{L}$ operator.

For $\alpha \in [\![1, m]\!]$, we define Γ_α^D and Γ_α^R as open subsets of Γ, possibly empty, such that $\Gamma_\alpha^D \cap \Gamma_\alpha^R = \emptyset$. Let $\boldsymbol{f} \in (L^2(\Omega))^m$, $g_\alpha^D \in \mathrm{H}^{1/2}(\Gamma_\alpha^D)$, $g_\alpha^R \in L^2(\Gamma_\alpha^R)$, $a_\alpha^R \in L^\infty(\Gamma_\alpha^R)$ be given data.

A *vector* boundary value problem is given by

Vector BVP 1 : generic problem

Find $\boldsymbol{u} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m) \in (\mathrm{H}^2(\Omega))^m$ such that

$$\mathcal{H}(\boldsymbol{u}) = \boldsymbol{f} \quad \text{in } \Omega, \tag{1.14}$$

$$\boldsymbol{u}_\alpha = g_\alpha^D \quad \text{on } \Gamma_\alpha^D, \ \forall \alpha \in [\![1, m]\!], \tag{1.15}$$

$$\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}_\alpha}} + a_\alpha^R \boldsymbol{u}_\alpha = g_\alpha^R \quad \text{on } \Gamma_\alpha^R, \ \forall \alpha \in [\![1, m]\!], \tag{1.16}$$

where the α-th component of the **conormal derivative** of $\boldsymbol{u}$ is defined by

$$\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}_\alpha}} \stackrel{\text{def}}{=} \sum_{\beta=1}^{m} \frac{\partial \boldsymbol{u}_\beta}{\partial n_{\mathcal{H}_{\alpha,\beta}}} = \sum_{\beta=1}^{m} \left(\left\langle \mathbb{A}^{\alpha,\beta} \boldsymbol{\nabla} \boldsymbol{u}_\beta, \boldsymbol{n} \right\rangle - \left\langle \boldsymbol{b}^{\alpha,\beta} \boldsymbol{u}_\beta, \boldsymbol{n} \right\rangle \right). \tag{1.17}$$

The boundary conditions (1.16) are the **Robin** boundary conditions and (1.15) is the **Dirichlet** boundary condition. The **Neumann** boundary conditions are particular Robin boundary conditions with $a_\alpha^R \equiv 0$.

In this problem, we may consider on a given boundary some conditions which can vary depending on the component. For example we may have a Robin boundary condition satisfying $\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}_1}} + a_1^R \boldsymbol{u}_1 = g_1^R$ and a Dirichlet one with $\boldsymbol{u}_2 = g_2^D$.

To have an outline of the FC-VFEM$\mathbb{P}_1$ package, a second and simple problem is quickly present.

Usual vector BVP 1 : 2D simple vector problem

Find $\boldsymbol{u} = (u_1, u_2) \in (\mathrm{H}^2(\Omega))^2$ such that

$$-\Delta u_1 + u_2 = 0 \text{ in } \Omega \subset \mathbb{R}^2, \quad (1.18)$$
$$-\Delta u_2 + u_1 = 0 \text{ in } \Omega \subset \mathbb{R}^2, \quad (1.19)$$
$$(u_1, u_2) = (0, 0) \text{ on } \Gamma_1, \quad (1.20)$$
$$(u_1, u_2) = (-12., +12.) \text{ on } \Gamma_{98}, \quad (1.21)$$
$$(u_1, u_2) = (+12., -12.) \text{ on } \Gamma_{99}, \quad (1.22)$$

where Ω and its boundaries are given in Figure 1.1.
The problem (1.18)-(1.22) can be equivalently expressed as the vector BVP (1.2)-(1.4) :

Vector BVP 2 : 2D simple vector problem

Find $\boldsymbol{u} = (u_1, u_2) \in (\mathrm{H}^2(\Omega))^2$ such that

$$\mathcal{H}(\boldsymbol{u}) = \boldsymbol{f} \quad \text{in } \Omega,$$
$$u_1 = g_1^D \quad \text{on } \Gamma^D = \Gamma_1 \cup \Gamma_{98} \cup \Gamma_{99},$$
$$u_2 = g_2^D \quad \text{on } \Gamma^D = \Gamma_1 \cup \Gamma_{98} \cup \Gamma_{99},$$

where

$$\mathcal{H} := \begin{pmatrix} \mathcal{L}_{\mathbb{I},\boldsymbol{O},\boldsymbol{O},0} & \mathcal{L}_{\mathbb{O},\boldsymbol{O},\boldsymbol{O},1} \\ \mathcal{L}_{\mathbb{O},\boldsymbol{O},\boldsymbol{O},1} & \mathcal{L}_{\mathbb{I},\boldsymbol{O},\boldsymbol{O},0} \end{pmatrix}, \text{ as } \mathcal{H}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -\Delta & 1 \\ 1 & -\Delta \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$
$$f \equiv 0,$$

and

$$g_1^D = g_2^D := 0 \text{ on } \Gamma_1,\ g_1^D := -12,\ g_2^D := +12 \text{ on } \Gamma_{98},\ g_1^D := +12,\ g_2^D := -12 \text{ on } \Gamma_{99}$$

In Listing 21 a complete code is given to solve this problem. Numerical solutions are given in Figure 1.2.

```
meshfile=gmsh.buildmesh2d('condenser',10); % generate mesh
Th=siMesh(meshfile); % read mesh
Hop=Hoperator(2,2,2);
Hop.set([1,2],[1,2],Loperator(2,2,{1,[];[],1},[],[],[]));
Hop.set([1,2],[2,1],Loperator(2,2,[],[],[],1));
pde=PDEelt(Hop);
bvp=BVP(Th,pde);
bvp.setDirichlet( 1, 0.,1:2);
bvp.setDirichlet( 98, {-12,+12},1:2);
bvp.setDirichlet( 99, {+12,-12},1:2);
U=bvp.solve('split',true);
% Graphic parts
figure(1)
Th.plot(U{1})
axis image;axis off;shading interp
colorbar
figure(2);
Th.plot(U{2})
axis image;axis off;shading interp
colorbar
```

Listing 1.2: Complete Octave code to solve the funny 2D vector problem with graphical representations

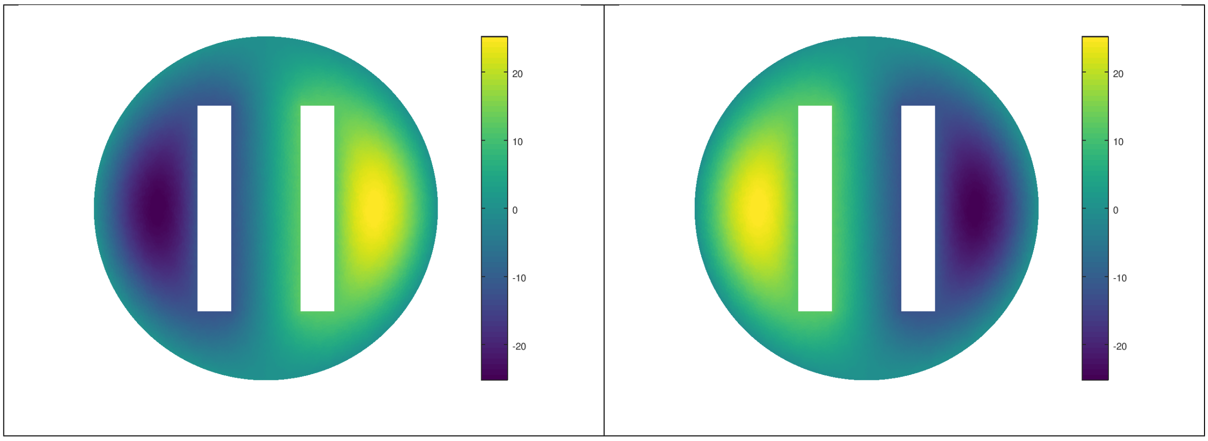


Figure 1.2: Funny vector BVP, u_1 numerical solution (left) and u_2 numerical solution (right)

Obviously, more complex problems will be studied in section **??** and complete explanations on the code will be given in next sections.

In the following of the report we will solve by a $\mathbb{P}_1$-Lagrange finite element method *scalar* B.V.P. (1.2) to (1.4) and *vector* B.V.P. (1.14) to (1.16) without additional restrictive assumption.

Chapter 2

Octave objects

2.1 Fdata object

This object is used to create the datas associated with the scalar boundary value problem (1.2)-(1.4) or vector boundary value problem (1.14)-(1.16).

2.2 Loperator object

The object Loperator is used to create the operator $\mathcal{L}_{\mathbb{A},\boldsymbol{b},\boldsymbol{c},a_0}$ defined in (1.1). Its main properties are

Properties of Loperator object

d	:	integer, space dimension.
A	:	array of d-by-d cells. Used to store the $\mathbb{A}$ functions such that A{i,j} $\leftarrow \mathbb{A}_{i,j}$. Each cell contains a Fdata object or is empty for 0 value.
b	:	array of d-by-1 cells. Used to store the $\boldsymbol{b}$ functions such that b{i} $\leftarrow \boldsymbol{b}_i$. Each cell contains a Fdata object or is empty for 0 value.
c	:	array of d-by-1 cells. Used to store the $\boldsymbol{c}$ functions such that c{i} $\leftarrow \boldsymbol{c}_i$. Each cell contains a Fdata object or is empty for 0 value.
a0	:	a Fdata object or empty for 0 value Used to store the a_0 function such that a0 $\leftarrow a_0$.
order	:	integer order of the operator : 2 if A is not empty, 1 if A is empty and b or c not empty, 0 if A, b and c are empty.

2.2.1 Constructor

Its contructor are

```
obj=Loperator()
obj=Lopertor(dim,d,A,b,c,a0)
```

Description

obj=Loperator() create an empty operator.

obj=Loperator(dim,d,A,b,c,a0) ...

-
-
-

Samples

$-\Delta u := \mathcal{L}_{\mathbb{I},\boldsymbol{O},\boldsymbol{O},0}$

in $\mathbb{R}$ Lop=Loperator(1,1,{1},[],[],[])
in $\mathbb{R}^2$ Lop=Loperator(2,2,{1,[];[],1},[],[],[])
in $\mathbb{R}^3$ Lop=Loperator(3,3,{1,[],[];[],1,[];[],[],1},[],[],[])
⋮

$-\Delta u + u := \mathcal{L}_{\mathbb{I},\boldsymbol{O},\boldsymbol{O},1}$

in $\mathbb{R}$ Lop=Loperator(1,1,{1},[],[],1)
in $\mathbb{R}^2$ Lop=Loperator(2,2,{1,[];[],1},[],[],1)
in $\mathbb{R}^3$ Lop=Loperator(3,3,{1,[],[];[],1,[];[],[],1},[],[],1)
⋮

In R^2, $-\Delta u + (1 + cos(x + y))u := \mathcal{L}_{\mathbb{I},\boldsymbol{O},\boldsymbol{O},(x,y)\mapsto(1+cos(x+y))}$

Lop=Loperator(2,2,{1,[];[],1},[],[],@(x,y)1+cos(x+y))

2.2.2 Methods

apply function

2.3 Hoperator object

The object Hoperator is used to create the operator $\mathcal{H}$ defined in (1.10). Its main properties are

Properties of Hoperator object

d	:	integer, space dimension.
m	:	integer
H	:	array of d-by-d cells. Used to store the $\mathcal{H}$ operators such that H{i,j} $\leftarrow \mathcal{H}_{i,j}$, $\forall i, j \in [\![1, \text{m}]\!]$. Each cell contains a Loperator object or an empty value.

2.3.1 Constructor

Its contructor are

```
obj=Hoperator()
obj=Hoperator(d,s,m)
```

Description

`obj=Hoperator()` create an empty operator with all dimensions set to 0.

`obj=Hopertor(d,s,m)` create an empty/null operator with the given dimensions.

-
-
-

Samples

In $\mathbb{R}^2$, with $\boldsymbol{u} = (u_1, u_2)$ the operator $\mathcal{H}$ defined by

$$\mathcal{H}(\boldsymbol{u}) \stackrel{\text{def}}{=} \begin{pmatrix} -\Delta u_1 + u_2 \\ u_1 - \Delta u_2 \end{pmatrix}$$

could be written as

$$\mathcal{H}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -\Delta & 1 \\ 1 & -\Delta \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

and then

$$\mathcal{H} = \begin{pmatrix} \mathcal{L}_{\mathbb{I},\boldsymbol{O},\boldsymbol{O},0} & \mathcal{L}_{\mathbb{O},\boldsymbol{O},\boldsymbol{O},1} \\ \mathcal{L}_{\mathbb{O},\boldsymbol{O},\boldsymbol{O},1} & \mathcal{L}_{\mathbb{I},\boldsymbol{O},\boldsymbol{O},0} \end{pmatrix}$$

```
Hop=Hoperator(2,2,2);
Lop1=Loperator(2,2,{1,[];[],1},[],[],[]);
Lop2=Loperator(2,2,[],[],[],1);
Hop.set(1,1,Lop1);Hop.set(2,2,Lop1);
Hop.set(1,2,Lop2);Hop.set(2,1,Lop2);
```

or

```
Hop=Hoperator(2,2,2);
Hop.set([1,2],[1,2],Loperator(2,2,{1,[];[],1},[],[],[]));
Hop.set([1,2],[2,1],Loperator(2,2,[],[],[],1));
```

2.3.2 Methods

set function

zeros function

opStiffElas function

2.4 PDEelt object

This object is used to create the scalar PDE (1.2) or the vector PDE (1.14):

$$\mathcal{L}(u) = f \quad \text{or} \quad \mathcal{H}(\boldsymbol{u}) = \boldsymbol{f}.$$

Its main properties are

Properties of PDEelt object

d	:	integer, space dimension.
m	:	integer
Op	:	Loperator or Hoperator object.
f	:	(cells of) Fdata object or empty. Used to store the right-hand side of the PDE. If Op is an Loperator object then f is an Fdata object or is empty. If Op is an Hoperator object then f is a cell array of Op.m Fdata object or empty value.

Its contructor are

```
obj=PDEelt()
obj=PDEelt(Op)
obj=PDEelt(Op,f)
```

Description

obj=PDEelt() create an empty object.

obj=PDEelt(Op) create the PDE with $f \equiv 0$: i.e. `Op(u)=0`

obj=PDEelt(Op,f) create the PDE `Op(u)=f`. If Op is an Hoperator object then f must be a cell array of length Hoperator.m.

Samples

In $\mathbb{R}^2$, $-\Delta u + u = f$, with $f(x,y) = x\sin(x+y)$

```
Lop=Loperator(2,2,{1,[];[],1},[],[],1);
f=@(x,y) x.*sin(x+y);
pde=PDEelt(Lop,f);
```

The f function must be written in a vectorized form.

2.5 BVP object

The object BVP is used to create a scalar boundary value problem (1.2)-(1.4) or a vector boundary value problem (1.14)-(1.16). The usage of this object is strongly correlated with good comprehension of the FC-SIMESH package and and more particularly with the siMesh object.

The properties of the object BVP are

Properties of BVP object

d	:	integer, space dimension.
m	:	integer, system of m PDEs.
Th	:	a siMesh object
pdes	:	Th.nsTh-by-1 cell array. Used to store the PDE associated with each submesh Th.sTh{i}. If pdes{i} is empty then there is no PDE defined on Th.sTh{i}.

2.5.1 Constructor

Its contructor are

```
obj=BVP()
obj=BVP(Th,pde)
obj=BVP(Th,pde,labels)
```

Description

`obj=BVP()` create an empty BVP object.

`obj=BVP(Th,pde)` create a BVP object with PDE's defined by pde object on all submeshes of index Th.find(pde.d) i.e. on all submeshes such that Th.sTh{i}==pde.d. By default, homogeneous Neumann boundary conditions are set on all *boundaries*.

`obj=BVP(Th,pde,labels)` similar to previous one except among the selected objects are choosen those with label (Th.sTh{i}.label) in labels array. By default, homogeneous Neumann boundary conditions are set on all *boundaries*.

2.5.2 Main methods

Let bvp be a BVP object.

setPDE function

```
bvp.setPDE(d,label,pde)
```

Description

`bvp.setPDE(d,label,pde)` associated the pde object with the i-th submesh such that i=bvp.Th.find(d,label) If i exists then bvp.pdes{i} is set to pde.

setDirichlet function

```
bvp.setDirichlet(label,g)
bvp.setDirichlet(label,g,Lm)
```

Description

`bvp.setDirichlet(label,g)` for scalar B.V.P., sets Dirichlet boundary condition

$$u = \mathrm{g}, \quad \text{on } \Gamma_{\text{label}}$$

and for vector B.V.P., sets Dirichlet boundary condition

$$u_i = \mathrm{g}\{\mathrm{i}\}, \forall i \in [\![1, m]\!] \text{ on } \Gamma_{\text{label}}.$$

`bvp.setDirichlet(label,g,Lm)` for vector B.V.P., sets Dirichlet boundary condition

$$u_{\mathrm{Lm(i)}} = \mathrm{g}\{\mathrm{i}\}, \forall i \in [\![1, \mathrm{length}(\mathrm{Lm})]\!] \text{ on } \Gamma_{\text{label}}.$$

setRobin function

```
bvp.setRobin(label,gr,ar)
bvp.setRobin(label,gr,ar,Lm)
```

Description

bvp.setRobin(label,gr,ar) for scalar B.V.P., sets Robin boundary condition (1.4)

$$\frac{\partial u}{\partial n_{\mathcal{L}}} + \mathrm{ar} u = \mathrm{gr}, \quad \text{on } \Gamma_{\mathrm{label}}.$$

For vector B.V.P., sets Robin boundary condition (1.16)

$$\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}_i}} + \mathrm{ar}\{\mathrm{i}\}\boldsymbol{u}_i = \mathrm{gr}\{\mathrm{i}\}, \quad \forall i \in [\![1, m]\!] \text{ on } \Gamma_{\mathrm{label}}.$$

bvp.setRobin(label,gr,ar,Lm) for vector B.V.P., sets Robin boundary condition (1.16) :
$\forall i \in [\![1, \mathrm{length}(\mathrm{Lm})]\!]$, let $\alpha = \mathrm{Lm}(\mathrm{i})$ then

$$\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}_\alpha}} + \mathrm{ar}\{\mathrm{i}\}\boldsymbol{u}_\alpha = \mathrm{gr}\{\mathrm{i}\}, \text{ on } \Gamma_{\mathrm{label}}.$$

solve function

```
x=bvp.solve()
x=bvp.solve(key,value,...)
```

Description

x=bvp.solve()) uses P_1-Lagrange finite elements method to solve the B.V.P. described by the bvp object.

x=bvp.solve(key,value,...)

- 'solver' :
- 'split' :
- 'local' :
- 'perm' :

Chapter 3

Scalar boundary value problems

3.1 Poisson BVP's

The generic problem to solve is the following

Usual BVP 2 : Poisson problem

Find $u \in \mathrm{H}^1(\Omega)$ such that

$$-\Delta u = f \quad \text{in } \Omega \subset \mathbb{R}^{\dim}, \tag{3.1}$$

$$u = g_D \quad \text{on } \Gamma_D, \tag{3.2}$$

$$\frac{\partial u}{\partial n} + a_R\, u = g_R \quad \text{on } \Gamma_R, \tag{3.3}$$

where $\Omega \subset \mathbb{R}^{\dim}$ with $\partial\Omega = \Gamma_D \cup \Gamma_R$ and $\Gamma_D \cap \Gamma_R = \emptyset$.

The Laplacian operator Δ can be rewritten according to a $\mathcal{L}$ operator defined in (1.1) and we have

$$-\Delta \stackrel{\mathsf{def}}{=} -\sum_{i=1}^{\dim} \frac{\partial^2}{\partial x_i{}^2} = \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}. \tag{3.4}$$

The conormal derivative $\frac{\partial u}{\partial n_{\mathcal{L}}}$ of this $\mathcal{L}$ operator is given by

$$\frac{\partial u}{\partial n_{\mathcal{L}}} \stackrel{\mathsf{def}}{=} \langle \mathbb{A}\, \boldsymbol{\nabla}\, u, \boldsymbol{n}\rangle - \langle \boldsymbol{b} u, \boldsymbol{n}\rangle = \frac{\partial u}{\partial n}. \tag{3.5}$$

We now will see how to implement different Poisson's BVP while using the FC-VFEM$\mathbb{P}_1$ toolbox.

3.1.1 2D Poisson BVP with Dirichlet boundary conditions on the unit square

Let Ω be the unit square with the associated mesh obtain from HYPERCUBE function (see section ?? for explanation and Figure ?? for a mesh sample) by the command

```
Th=fc_simesh.HyperCube(2,50);
```

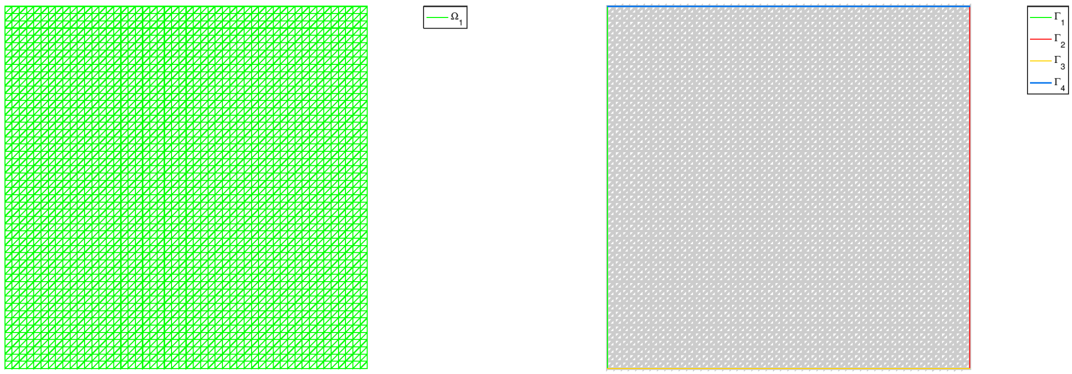


Figure 3.1: 2D hypercube (left) and its boundaries (right)

We choose the problem to have exact solution

$$u_{\text{ex}}(x, y) = \cos(x - y)\sin(x + y) + e^{\left(-x^2 - y^2\right)}.$$

So we set $f = -\Delta u_{\text{ex}}$ i.e.

$$f(x, y) = -4\,x^2 e^{\left(-x^2-y^2\right)} - 4\,y^2 e^{\left(-x^2-y^2\right)} + 4\,\cos(x - y)\sin(x + y) + 4\,e^{\left(-x^2-y^2\right)}.$$

On all the 4 boundaries we set a Dirichlet boundary conditions (and so $\Gamma_R = \emptyset$) :

$$u = u_{\text{ex}}, \text{ on } \Gamma_D = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4.$$

So this problem can be written as the scalar BVP 5

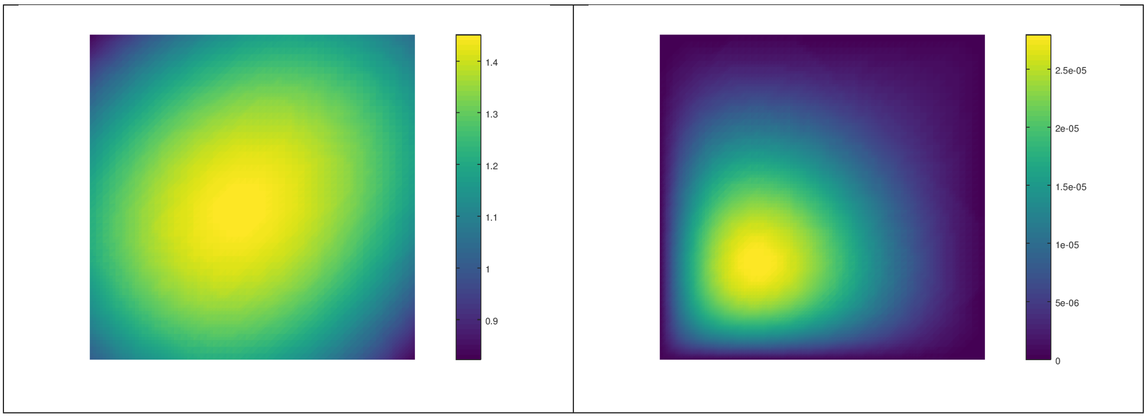
***Scalar* BVP 3 : 2D Poisson BVP with Dirichlet boundary conditions**

Find $u \in \mathrm{H}^1(\Omega)$ such that

$$\mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}(u) = f \text{ in } \Omega = [0,1]^2, \quad (3.6)$$

$$u = u_{\text{ex}} \text{ on } \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4, \quad (3.7)$$

In Listing 9, we give the complete code to solve this problem with FC-VFEM$\mathbb{P}_1$ toolbox.




```
uex=@(x,y) cos(x-y).*sin(x+y)+exp(-(x.^2+y.^2));
f=@(x,y) -4*x.^2.*exp(-x.^2-y.^2) - 4*y.^2.*exp(-x.^2-y.^2) + 4*cos(x-y).*sin(x+y) + ...
    4*exp(-x.^2-y.^2);
Th=fc_simesh.HyperCube(2,50);
Lop=Loperator(2,2,{1,0;0,1},[],[],[]);
pde=PDEelt(Lop,f);
bvp=BVP(Th,pde);
for lab=1:4, bvp.setDirichlet( lab, uex);end % Setting Dirichlet boundary conditions
U=bvp.solve(); % Solving the BVP
```

Listing 3.1: Poisson 2D BVP with Dirichlet boundary conditions : numerical solution (left) and error (right)

In line **??** we set the Dirichlet boundary conditions and in line **??** we solve the BVP.

3.1.2 2D Poisson BVP with mixed boundary conditions

Let Ω be the unit square with the associated mesh obtain from HyperCube function (see section **??** for explanation and Figure **??** for a mesh sample)

We choose the problem to have exact solution

$$u_{\text{ex}}(x, y) = \cos(2x + y).$$

So we set $f = -\Delta u_{\text{ex}}$ i.e.

$$f(x, y) = 5\cos(2x + y).$$

On boundary labels 1 and 2 we set a Dirichlet boundary conditions :

$$u = u_{\text{ex}}, \text{ on } \Gamma^D = \Gamma_1 \cup \Gamma_2.$$

On boundary label 3, we choose a Robin boundary condition with $a^R(x, y) = x^2 + y^2 + 1$. So we have

$$\frac{\partial u}{\partial n} + a^R u = g^R, \text{ on } \Gamma^R = \Gamma_3$$

with $g^R = (x^2 + y^2 + 1)\cos(2x + y) + \sin(2x + y)$.

On boundary label 4, we choose a Newmann boundary condition. So we have

$$\frac{\partial u}{\partial n} = g^N, \text{ on } \Gamma^N = \Gamma_4$$

with $g^N = -\sin(2x + y)$. this can be also written in the form of a Robin condition with $aR = 0$

So this problem can be written as the scalar BVP 5

Scalar BVP 4 : 2D Poisson BVP with Dirichlet boundary conditions

Find $u \in \mathrm{H}^1(\Omega)$ such that

$$\mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}(u) = f \quad \text{in } \Omega = [0,1]^2, \tag{3.8}$$

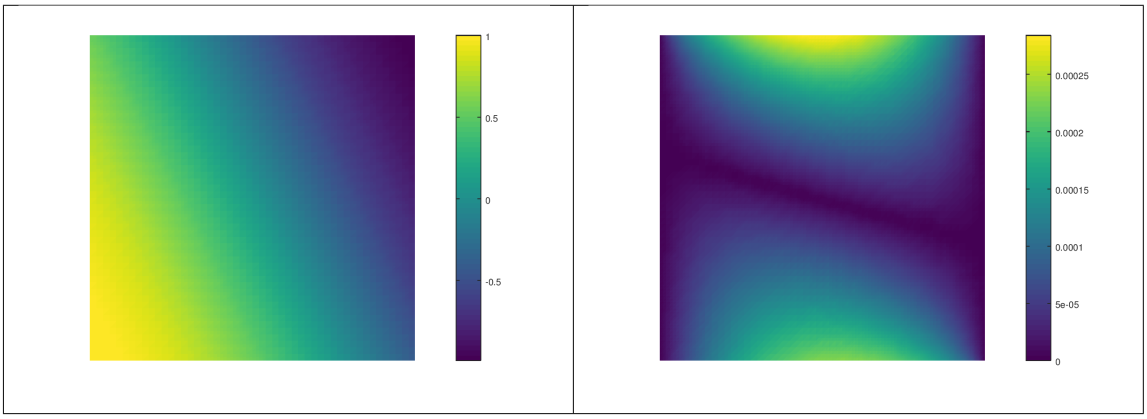
$$u = u_{\text{ex}} \quad \text{on } \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4, \tag{3.9}$$

$$\frac{\partial u}{\partial n_{\mathcal{L}}} + a^R u = g^R \quad \text{on } \Gamma_3, \tag{3.10}$$

$$\frac{\partial u}{\partial n_{\mathcal{L}}} = g^N \quad \text{on } \Gamma_4, \tag{3.11}$$

$$\tag{3.12}$$

In Listing 14, we give the complete code to solve this problem with FC-VFEM$\mathbb{P}_1$ toolbox.




```
1  uex=@(x,y) cos(2*x+y);
2  f=@(x,y) 5*cos(2*x+y);
3  gradu={@(x,y) -2*sin(2*x+y), @(x,y) -sin(2*x+y)};
4  ar3=@(x,y) 1+x.^2+y.^2;
5  Th=fc_simesh.HyperCube(2,50);
6  Lop=Loperator(2,2,{1,0;0,1},[],[],[]);
7  pde=PDEelt(Lop,f);
8  bvp=BVP(Th,pde);
9  bvp.setDirichlet( 1, uex);
10 bvp.setDirichlet( 2, uex);
11 bvp.setRobin( 3, @(x,y) -gradu{2}(x,y)+ar3(x,y).*uex(x,y),ar3);
12 bvp.setRobin( 4, gradu{2},[]);
13 U=bvp.solve();
```

Listing 3.2: Poisson 2D BVP with mixed boundary conditions : numerical solution (left) and error (right)

We set respectively in lines 11 and 12, the Robin and the Neumann boundary conditions by using SETROBIN member function of `BVP` class.

3.1.3 3D Poisson BVP with mixed boundary conditions

Let Ω be the unit cube with the associated mesh obtain from HYPERCUBE function (see section **??** for explanation and Figure **??** for a mesh sample)

We choose the problem to have exact solution

$$u_{\text{ex}}(x, y, y) = \cos(4\,x - 3\,y + 5\,z).$$

So we set $f = -\Delta u_{\text{ex}}$ i.e.

$$f(x, y, z) = 50\,\cos(4\,x - 3\,y + 5\,z).$$

On boundary labels $1, 3, 5$ we set a Dirichlet boundary conditions :

$$u = u_{\text{ex}}, \ \text{on } \Gamma^D = \Gamma_1 \cup \Gamma_3 \cup \Gamma_5.$$

On boundary label 2, we choose a Robin boundary condition with $a^R(x,y) = 1$. So we have

$$\frac{\partial u}{\partial n} + a^R u = g^R, \text{ on } \Gamma^R = \Gamma_2 \cup \Gamma_4$$

with $g^R(x,y,z) = \cos(4\,x - 3\,y + 5\,z) - 4\,\sin(4\,x - 3\,y + 5\,z)$, on Γ_2 and $g^R(x,y,z) = \cos(4\,x - 3\,y + 5\,z) + 3\,\sin(4\,x - 3\,y + 5\,z)$, on Γ_4.

On boundary label 6, we choose a Newmann boundary condition. So we have

$$\frac{\partial u}{\partial n} = g^N, \text{ on } \Gamma^N = \Gamma_6$$

with $g^N = -5\,\sin(4\,x - 3\,y + 5\,z)$. this can be also written in the form of a Robin condition with $aR = 0$ on Γ_6.

So this problem can be written as the scalar BVP 5

Scalar BVP 5 : 3D Poisson BVP with mixed boundary conditions

Find $u \in \mathrm{H}^1(\Omega)$ such that

$$\mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}(u) = f \quad \text{in } \Omega = [0,1]^3, \tag{3.13}$$

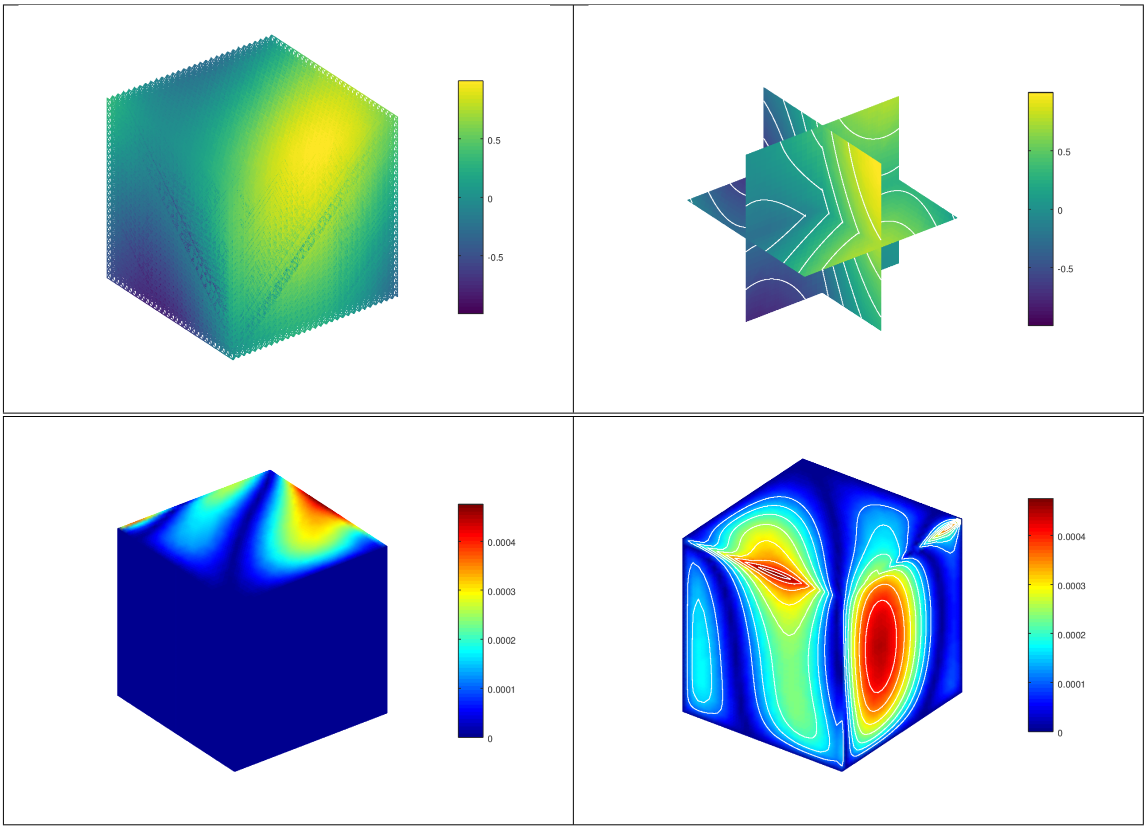
$$u = u_{\text{ex}} \quad \text{on } \Gamma_1 \cup \Gamma_3 \cup \Gamma_5, \tag{3.14}$$

$$\frac{\partial u}{\partial n_{\mathcal{L}}} + a^R u = g^R \quad \text{on } \Gamma_2 \cup \Gamma_4, \tag{3.15}$$

$$\frac{\partial u}{\partial n_{\mathcal{L}}} = g^N \quad \text{on } \Gamma_6, \tag{3.16}$$

(3.17)

In Listing 16, we give the complete code to solve this problem with FC-VFEM$\mathbb{P}_1$ toolbox.




```
uex=@(x,y,z) cos(2∗x−y−z).∗sin(x−2∗y+z);
f=@(x,y,z) 6∗cos(x − 2∗y + z).∗sin(2∗x − y − z) + 12∗cos(2∗x − y − z).∗sin(x − 2∗y + z);
ar=1;
gradu={@(x,y,z) cos(2∗x − y − z).∗cos(x − 2∗y + z) − 2∗sin(2∗x − y − z).∗sin(x − 2∗y + z), ...
        @(x,y,z) −2∗cos(2∗x − y − z).∗cos(x − 2∗y + z) + sin(2∗x − y − z).∗sin(x − 2∗y + z), ...
        @(x,y,z) cos(2∗x − y − z).∗cos(x − 2∗y + z) + sin(2∗x − y − z).∗sin(x − 2∗y + z)};
Th=fc_simesh.HyperCube(3,30);
Lop=Loperator(3,3,{1,0,0;0,1,0;0,0,1},[],[],[]);
pde=PDEelt(Lop,f);
bvp=BVP(Th,pde);
for lab=[1,3,5], bvp.setDirichlet( lab, uex);end
bvp.setRobin(2,@(x,y,z) gradu{1}(x,y,z)+ar∗uex(x,y,z),ar);
bvp.setRobin(4,@(x,y,z) gradu{2}(x,y,z)+ar∗uex(x,y,z),ar);
bvp.setRobin(6,@(x,y,z) gradu{3}(x,y,z),[]);
U=bvp.solve();
```

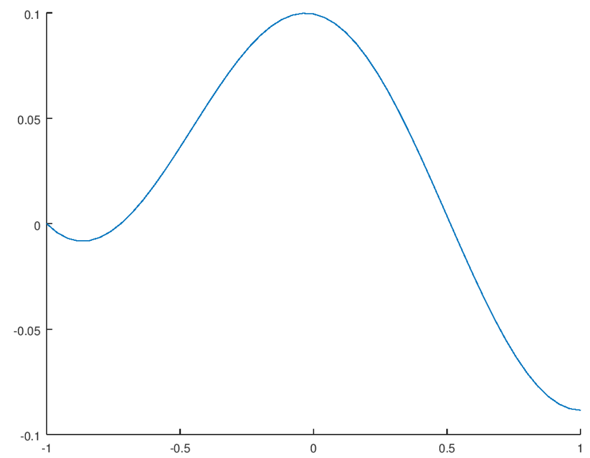
Listing 3.3: 3D Poisson BVP with mixed boundary conditions : numerical solution (upper) and error (bottom)

3.1.4 1D BVP : just for fun

Let Ω be the interval $[a, b]$ we want to solve the following PDE

$$-u''(x) + c(x)u(x) = f(x) \quad \forall x \in]a, b[$$

with the Dirichlet boundary condition $u(a) = 0$ and the homgeneous Neumann boundary condition on b




```
f=@(x) cos(pi*x);
c=@(x) 1+(x-1).^2;
a=-1;b=1;
Th=fc_simesh.HyperCube(1,50,'trans',@(x) a + (b-a)*x);
Lop=Loperator(1,1,{1},[],[],c);
pde=PDEelt(Lop,f);
bvp=BVP(Th,pde);
bvp.setDirichlet( 1, 0);
U=bvp.solve();
```

Listing 3.4: 1D BVP with mixed boundary conditions

3.2 Stationary convection-diffusion problem

3.2.1 Stationary convection-diffusion problem in 2D

The 2D problem to solve is the following

Usual BVP 3 : 2D stationary convection-diffusion problem

Find $u \in \mathrm{H}^1(\Omega)$ such that

$$-\operatorname{div}(\alpha \boldsymbol{\nabla} u) + \langle \boldsymbol{V}, \boldsymbol{\nabla} u \rangle + \beta u = f \quad \text{in } \Omega \subset \mathbb{R}^2, \tag{3.18}$$

$$u = 4 \quad \text{on } \Gamma_2, \tag{3.19}$$

$$u = -4 \quad \text{on } \Gamma_4, \tag{3.20}$$

$$u = 0 \quad \text{on } \Gamma_{20} \cup \Gamma_{21}, \tag{3.21}$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma_1 \cup \Gamma_3 \cup \Gamma_{10} \tag{3.22}$$

where Ω and its boundaries are given in Figure **??**. This problem is well posed if $\alpha(\boldsymbol{x}) > 0$ and $\beta(\boldsymbol{x}) \geqslant 0$.
We choose α, $\boldsymbol{V}$, β and f in Ω as :

$$\begin{aligned}
\alpha(\boldsymbol{x}) &= 0.1 + (x_1 - 0.5)^2, \\
\boldsymbol{V}(\boldsymbol{x}) &= (-10x_2, 10x_1)^t, \\
\beta(\boldsymbol{x}) &= 0.01, \\
f(\boldsymbol{x}) &= -200\exp(-10((x_1 - 0.75)^2 + x_2^2)).
\end{aligned}$$

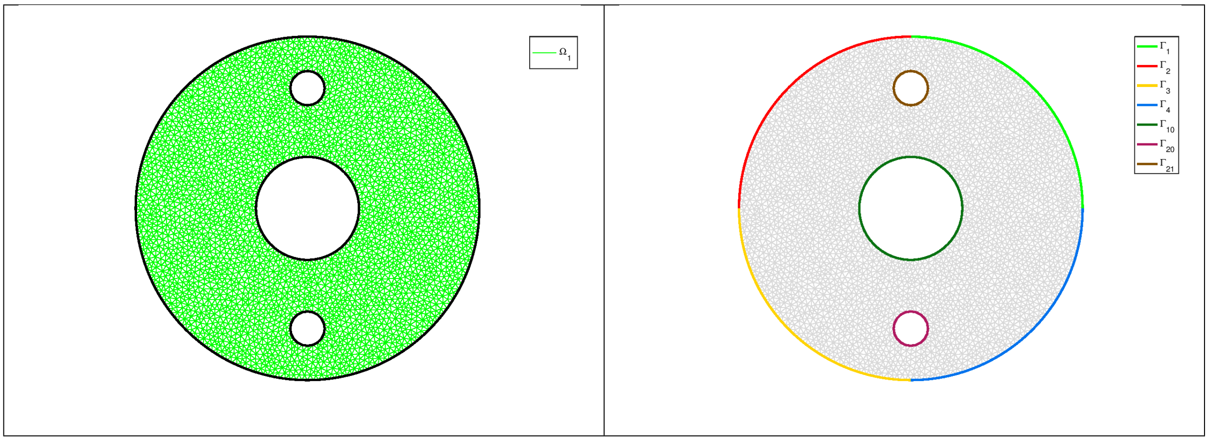


Figure 3.2: 2D stationary convection-diffusion BVP : mesh (left) and boundaries (right)

The problem (3.18)-(3.22) can be equivalently expressed as the scalar BVP (1.2)-(1.4) :

Scalar BVP 6 : 2D stationary convection-diffusion problem

Find $u \in \mathrm{H}^1(\Omega)$ such that

$$\begin{aligned} \mathcal{L}(u) &= f && \text{in } \Omega, \\ u &= g^D && \text{on } \Gamma^D, \\ \frac{\partial u}{\partial n_{\mathcal{L}}} + a^R u &= g^R && \text{on } \Gamma^R. \end{aligned}$$

where

- $\mathcal{L} := \mathcal{L}_{\alpha\mathbb{I},\mathbf{0},\boldsymbol{V},\beta}$, and then the conormal derivative of u is given by

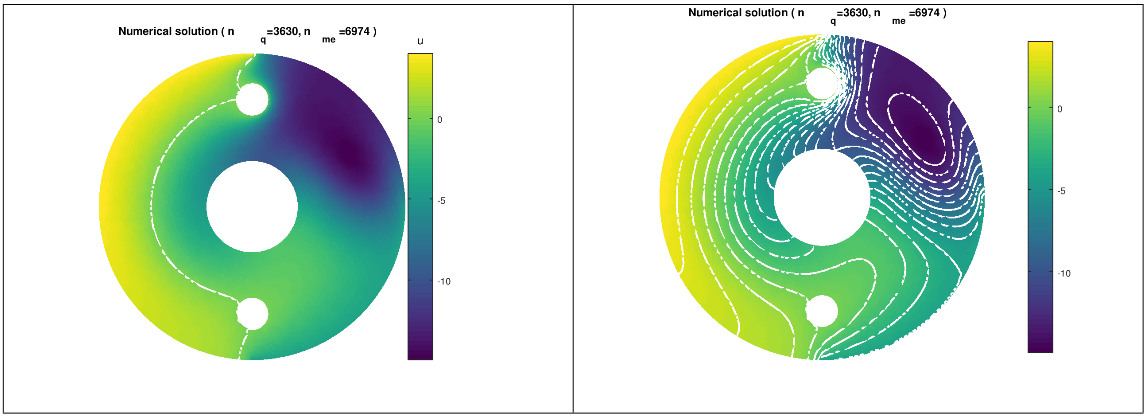
$$\frac{\partial u}{\partial n_{\mathcal{L}}} := \langle \mathbb{A}\boldsymbol{\nabla} u, \boldsymbol{n}\rangle - \langle \boldsymbol{b} u, \boldsymbol{n}\rangle = \alpha \frac{\partial u}{\partial n}.$$

- $\Gamma^D = \Gamma_2 \cup \Gamma_4 \cup \Gamma_{20} \cup \Gamma_{21}$ and $\Gamma^R = \Gamma_1 \cup \Gamma_3 \cup \Gamma_{10}$
- $g^D := 4$ on Γ_2, and $g^D := -4$ on Γ_4 and $g^D := 0$ on $\Gamma_{20} \cup \Gamma_{21}$
- $a^R = g^R := 0$ on Γ^R.

The algorithm using the toolbox for solving (3.18)-(3.22) is the following:

Algorithm 1 Stationary convection-diffusion problem in 2D

$\mathcal{T}_h \leftarrow$ siMesh(...) ▷ Get mesh
$\alpha \leftarrow (x, y) \longmapsto 0.1 + (y - 0.5)(y - 0.5)$
$\beta \leftarrow 0.01$
$f \leftarrow (x, y) \longmapsto -200e^{-10((x-0.75)^2+y^2)}$
Lop ← LOperator$\left(2, 2, \begin{pmatrix} \alpha & 0 \\ 0 & \alpha \end{pmatrix}, \mathbf{0}, \begin{pmatrix} -10y \\ 10x \end{pmatrix}, \beta\right)$
pde ← PDEelt(Lop, f)
bvp ← BVP($\mathcal{T}_h$, pde)
bvp.setDirichlet(2, 4.0) ▷ Set 'Dirichlet' condition on Γ_2
bvp.setDirichlet(4, −4.0) ▷ Set 'Dirichlet' condition on Γ_4
bvp.setDirichlet(20, 0.0) ▷ Set 'Dirichlet' condition on Γ_{20}
bvp.setDirichlet(21, 0.0) ▷ Set 'Dirichlet' condition on Γ_{21}
$\boldsymbol{u} \leftarrow$ bvp.solve()




```
fullgeofile=fc_vfemp1.get_geo(2,2,geofile);
if isempty(fullgeofile), error('geofile␣%s␣not␣found',geofile);end
af=@(x,y) 0.1+y.^2;
Vx=@(x,y) −10*y;Vy=@(x,y) 10*x;
meshfile=gmsh.buildmesh2d(fullgeofile,N,'meshdir',R.meshdir);%,'force',true);,'geodir',R.geodir
end
tstart=tic();
Lop=Loperator(Th.dim,Th.d,{af,[];[],af},[],{Vx,Vy},b);
pde=PDEelt(Lop,f);
bvp=BVP(Th,pde);
bvp.setDirichlet(2, g2);
bvp.setDirichlet(4, g4);
```

Listing 3.5: Setting the 2D stationary convection-diffusion BVP and representation of the numerical solution

The numerical solution for a given mesh is shown on figures of Listing **??**

3.2.2 Stationary convection-diffusion problem in 3D

Let $A = (x_A, y_A) \in \mathbb{R}^2$ and $\mathcal{C}_A^r([z_{min}, z_{max}])$ be the right circular cylinder along $z-$axis ($z \in [z_{min}, z_{max}]$) with bases the circles of radius r and center (x_A, y_A, z_{min}) and (x_A, y_A, z_{max}).

Let Ω be the cylinder defined by

$$\Omega = \mathcal{C}^1_{(0,0)}([0,3]) \backslash \{\mathcal{C}^{0.3}_{(0,0)}([0,3]) \cup \mathcal{C}^{0.1}_{(0,-0.7)}([0,3]) \cup \mathcal{C}^{0.1}_{(0,0.7)}([0,3])\}.$$

We respectively denote by Γ_{1000} and Γ_{1001} the $z = 0$ and $z = 3$ bases of Ω.

Γ_1, Γ_{10}, Γ_{20} and Γ_{21} are respectively the curved surfaces of cylinders $\mathcal{C}^1_{(0,0)}([0,3])$, $\mathcal{C}^{0.3}_{(0,0)}([0,3])$, $\mathcal{C}^{0.1}_{(0,-0.7)}([0,3])$ and $\mathcal{C}^{0.1}_{(0,0.7)}([0,3])$.

The domain Ω and its boundaries are represented in Figure **??**.

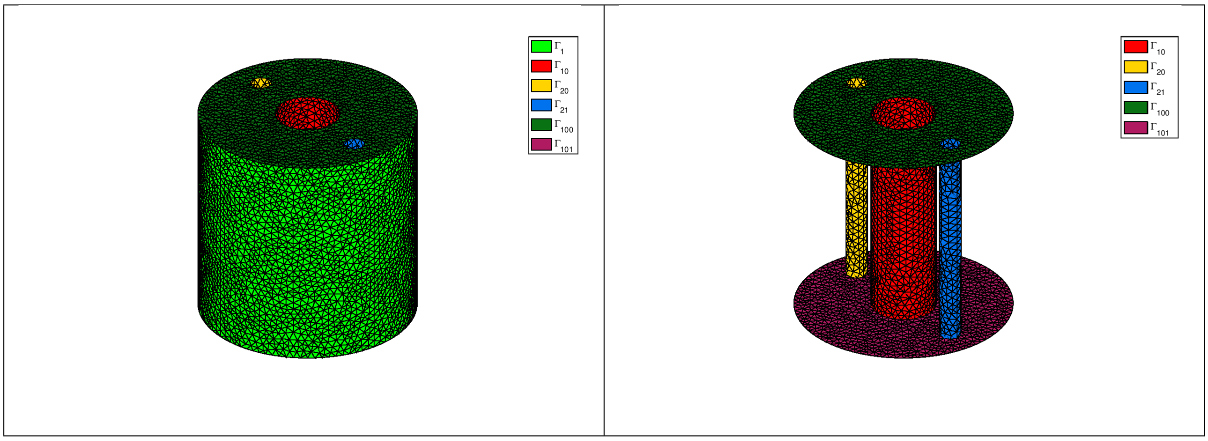


Figure 3.3: 3D stationary convection-diffusion BVP : all boundaries (left) and boundaries without Γ_1 (right)

The 3D problem to solve is the following

***Usual* BVP 4 :**

3D problem : Stationary convection-diffusion Find $u \in \mathrm{H}^2(\Omega)$ such that

$$-\operatorname{div}(\alpha \boldsymbol{\nabla} u) + \langle \boldsymbol{V}, \boldsymbol{\nabla} u \rangle + \beta u = f \quad \text{in } \Omega \subset \mathbb{R}^3, \tag{3.23}$$

$$\alpha \frac{\partial u}{\partial n} + a_{20} u = g_{20} \quad \text{on } \Gamma_{20}, \tag{3.24}$$

$$\alpha \frac{\partial u}{\partial n} + a_{21} u = g_{21} \quad \text{on } \Gamma_{21}, \tag{3.25}$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma^N \tag{3.26}$$

where $\Gamma^N = \Gamma_1 \cup \Gamma_{10} \cup \Gamma_{1000} \cup \Gamma_{1001}$. This problem is well posed if $\alpha(\boldsymbol{x}) > 0$ and $\beta(\boldsymbol{x}) \geqslant 0$. We choose $a_{20} = a_{21} = 1$, $g_{21} = -g_{20} = 0.05$ $\beta = 0.01$ and :

$$\begin{aligned}
\alpha(\boldsymbol{x}) &= 0.7 + \boldsymbol{x}_3/10, \\
\boldsymbol{V}(\boldsymbol{x}) &= (-10x_2, 10x_1, 10x_3)^t, \\
f(\boldsymbol{x}) &= -800 \exp(-10((x_1 - 0.65)^2 + x_2^2 + (x_3 - 0.5)^2)) \\
&\quad +800 \exp(-10((x_1 + 0.65)^2 + x_2^2 + (x_3 - 0.5)^2)).
\end{aligned}$$

The problem (3.23)-(3.26) can be equivalently expressed as the scalar BVP (1.2)-(1.4) :

***Scalar* BVP 7 :**

3D stationary convection-diffusion problem as a *scalar* BVP Find $u \in \mathrm{H}^2(\Omega)$ such that

$$\begin{aligned}
\mathcal{L}(u) &= f && \text{in } \Omega, \\
\frac{\partial u}{\partial n_{\mathcal{L}}} + a^R u &= g^R && \text{on } \Gamma^R.
\end{aligned}$$

where

- $\mathcal{L} := \mathcal{L}_{\alpha\mathbb{I}, \mathbf{0}, \boldsymbol{V}, \beta}$, and then the conormal derivative of u is given by

$$\frac{\partial u}{\partial n_{\mathcal{L}}} := \langle \mathbb{A} \boldsymbol{\nabla} u, \boldsymbol{n} \rangle - \langle \boldsymbol{b} u, \boldsymbol{n} \rangle = \alpha \frac{\partial u}{\partial n}.$$

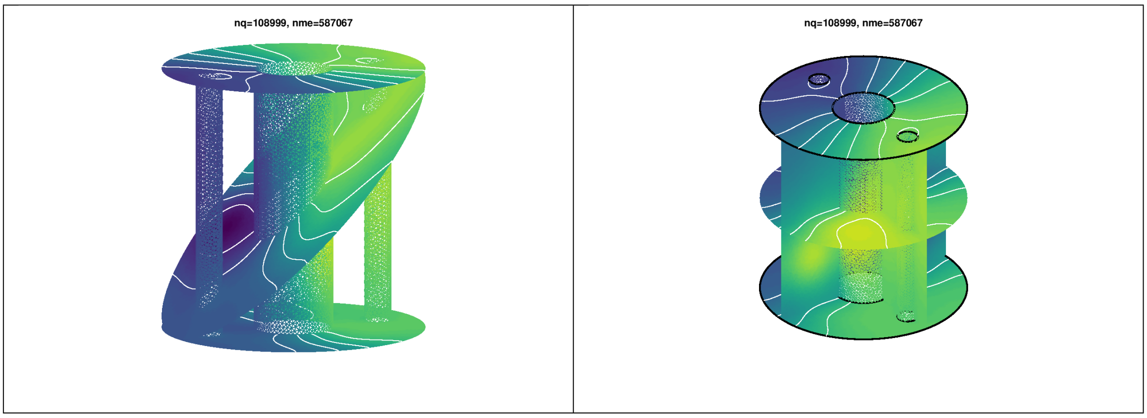
- $\Gamma^R = \Gamma_1 \cup \Gamma_{10} \cup \Gamma_{20} \cup \Gamma_{21} \cup \Gamma_{1000} \cup \Gamma_{1001}$ (and $\Gamma^D = \emptyset$)

•

$$a^R = \begin{cases} 0 & \text{on } \Gamma_1 \cup \Gamma_{10} \cup \Gamma_{1000} \cup \Gamma_{1001} \\ 1 & \text{on } \Gamma_{20} \cup \Gamma_{21} \end{cases}$$

$$g^R = \begin{cases} 0 & \text{on } \Gamma_1 \cup \Gamma_{10} \cup \Gamma_{1000} \cup \Gamma_{1001} \\ 0.05 & \text{on } \Gamma_{21}, \\ -0.05 & \text{on } \Gamma_{20} \end{cases}$$

We give respectively in Listing 11 the corresponding Octave codes and the numerical solution for a more refined mesh.




```
fullgeofile=fc_vfemp1.get_geo(3,3,geofile);
if isempty(fullgeofile), error('geofile␣%s␣not␣found',geofile);end
af=@(x,y,z) 0.7+ z/10;
beta=0.01;
tstart=tic();
end
tstart=tic();
Lop=Loperator(Th.dim,Th.d,{af,[],[];[],af,[];[],[],af},[],V,beta);
pde=PDEelt(Lop,f);
bvp=BVP(Th,pde);
```

Listing 3.6: Setting the 3D stationary convection-diffusion BVP and representation of the numerical solution

3.3 2D electrostatic BVPs

In this sample, we shall discuss electrostatic solutions for current flow in resistive media. Consider a region Ω of contiguous solid and/or liquid conductors. Let $\boldsymbol{j}$ be the current density in A/m^2. It's satisfy

$$\operatorname{div} \boldsymbol{j} = 0, \quad \text{in } \Omega. \tag{3.27}$$

$$\boldsymbol{j} = \sigma \boldsymbol{E}, \quad \text{in } \Omega. \tag{3.28}$$

where σ is the local electrical conductivity and $\boldsymbol{E}$ the local electric field.

The electric field can be written as a gradient of a scalar potential

$$\boldsymbol{E} = -\boldsymbol{\nabla} \varphi, \quad \text{in } \Omega. \tag{3.29}$$

Combining all these equations leads to Laplace's equation

$$\operatorname{div}(\sigma \boldsymbol{\nabla} \varphi) = 0 \tag{3.30}$$

In the resistive model, a good conductor has high value of σ and a good insulator has $0 < \sigma\mu 1$.

Material	$\rho(\Omega.m)$ at $20°C$	$\sigma(S/m)$ at $20°C$
Carbon (graphene)	1.00×10^{-8}	1.00×10^{8}
Gold	2.44×10^{-8}	4.10×10^{8}
Drinking water	2.00×10^{1} to 2.00×10^{3}	5.00×10^{-4} to 5.00×10^{-2}
Silicon	6.40×10^{2}	1.56×10^{-3}
Glass	1.00×10^{11} to 1.00×10^{15}	10^{-15} to 10^{-11}
Air	1.30×10^{16} to 3.30×10^{16}	3×10^{-15} to 8×10^{-15}

As example, we use the mesh obtain with `gmsh` from `square4holes6dom.geo` file represented in Figure 3.4

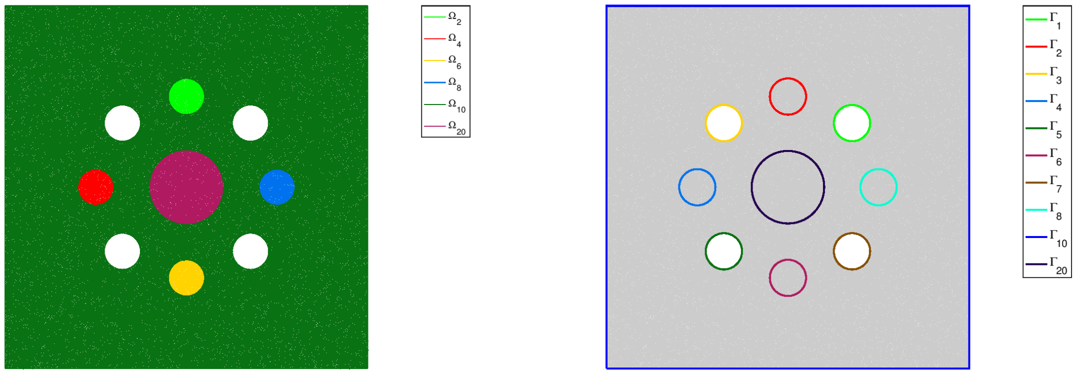


Figure 3.4: Mesh from `square4holes6dom.geo`, domains representation (left) and boundaries (right)

We have two resistive medias

$$\Omega_a = \Omega_{10} \quad \text{and} \quad \Omega_b = \Omega_{20} \cup \Omega_2 \cup \Omega_4 \cup \Omega_6 \cup \Omega_8.$$

In Ω_a and Ω_b the local electrical conductivity are respectively given by

$$\sigma = \begin{cases} \sigma_a = 10^4, & \text{in } \Omega_a \\ \sigma_b = 10^{-4} & \text{in } \Omega_a \end{cases}$$

We solve the following BVP

Usual BVP 5 : 2D electrostatic problem

Find $\varphi \in \mathrm{H}^1(\Omega)$ such that

$$\operatorname{div}(\sigma \nabla \varphi) = 0 \quad \text{in } \Omega, \tag{3.31}$$

$$\varphi = 0 \quad \text{on } \Gamma_3 \cup \Gamma_7, \tag{3.32}$$

$$\varphi = 12 \quad \text{on } \Gamma_1 \cup \Gamma_5, \tag{3.33}$$

$$\sigma \frac{\partial \varphi}{\partial n} = 0 \quad \text{on } \Gamma_{10}. \tag{3.34}$$

The problem (3.31)-(3.34) can be equivalently expressed as the scalar BVP (1.2)-(1.4) :

Scalar BVP 8 : 2D electrostatic problem

Find $\varphi \in \mathrm{H}^1(\Omega)$ such that

$$\mathcal{L}(\varphi) = 0 \quad \text{in } \Omega,$$

$$\varphi = g^D \quad \text{on } \Gamma^D,$$

$$\frac{\partial \varphi}{\partial n_{\mathcal{L}}} + a^R \varphi = g^R \quad \text{on } \Gamma^R.$$

where

- $\mathcal{L} := \mathcal{L}_{\sigma\mathbb{I},\mathbf{0},\boldsymbol{V},\beta}$, and then the conormal derivative of φ is given by

$$\frac{\partial\varphi}{\partial n_{\mathcal{L}}} := \langle \mathbb{A}\,\boldsymbol{\nabla}\,\varphi, \boldsymbol{n}\rangle - \langle \boldsymbol{b}\varphi, \boldsymbol{n}\rangle = \sigma\frac{\partial\varphi}{\partial n}.$$

- $\Gamma^D = \Gamma_1 \cup \Gamma_3 \cup \Gamma_5 \cup \Gamma_7$ and $\Gamma^R = \Gamma_{10}$. The other borders should not be used to specify boundary conditions: they do not intervene in the variational formulation and in the physical problem!
- $g^D := 0$ on $\Gamma_3 \cup \Gamma_7$, and $g^D := 12$ on $\Gamma_1 \cup \Gamma_5$.
- $a^R = g^R := 0$ on Γ^R.

To write this problem properly with FC-VFEM$\mathbb{P}_1$ toolbox, we split (3.31) in two parts

$$\begin{aligned} \operatorname{div}(\sigma_a \,\boldsymbol{\nabla}\,\varphi) &= 0 \qquad \text{in } \Omega_a \\ \operatorname{div}(\sigma_b \,\boldsymbol{\nabla}\,\varphi) &= 0 \qquad \text{in } \Omega_b \end{aligned}$$

and we set these PDEs on each domains. This is done in Matlab Listing 3.7.

Listing 3.7: **Setting the 2D electrostatic BVP, Matlab code**

```
tstart=tic();
end
tstart=tic();
Lop=Loperator(dim,d,{sigma2,0;0,sigma2},[],[],[]);
pde=PDEelt(Lop);
bvp=BVP(Th,pde);
Lop=Loperator(dim,d,{sigma1,0;0,sigma1},[],[],[]);
pde=PDEelt(Lop);
bvp.setPDE(2,10,pde);
bvp.setDirichlet( 1, 12);
bvp.setDirichlet( 3, 0);
```

We show in Figures 3.5 and 3.6 respectively the potential φ and the norm of the electric field $\boldsymbol{E}$.

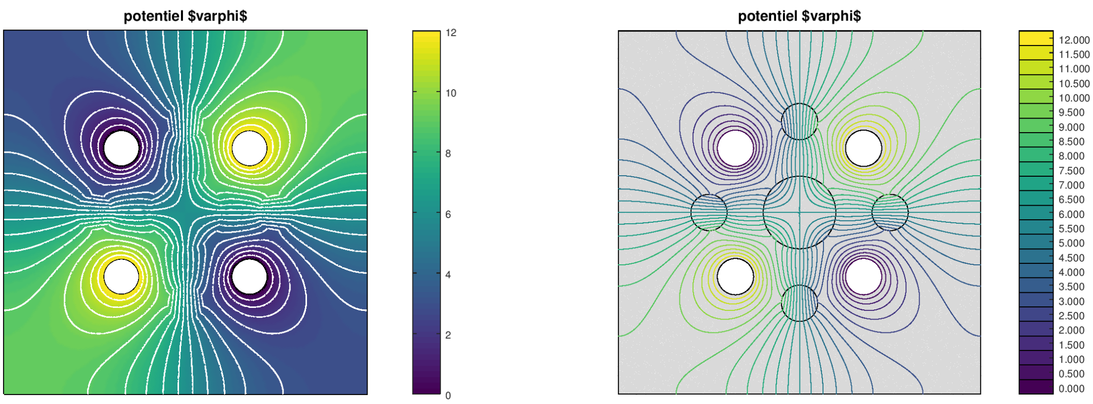


Figure 3.5: Test 1, potential φ

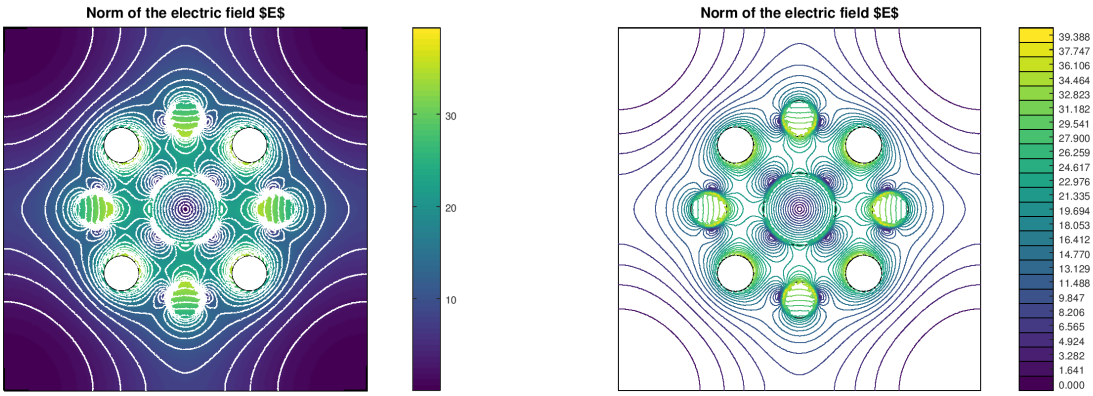


Figure 3.6: Test 1, norm of the electrical field $\boldsymbol{E}$

Chapter 4

Vector boundary value problems

4.1 Elasticity problem

4.1.1 General case ($d = 2, 3$)

We consider here Hooke's law in linear elasticity, under small strain hypothesis (see for example [3]).

For a sufficiently regular vector field $\boldsymbol{u} = (u_1, \ldots, u_d) : \Omega \to \mathbb{R}^d$, we define the linearized strain tensor $\underline{\boldsymbol{\epsilon}}$ by

$$\underline{\boldsymbol{\epsilon}}(\boldsymbol{u}) = \frac{1}{2}\left(\boldsymbol{\nabla}(\boldsymbol{u}) + \boldsymbol{\nabla}^t(\boldsymbol{u})\right).$$

We set $\underline{\boldsymbol{\epsilon}} = (\epsilon_{11}, \epsilon_{22}, 2\epsilon_{12})^t$ in 2d and $\underline{\boldsymbol{\epsilon}} = (\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, 2\epsilon_{12}, 2\epsilon_{23}, 2\epsilon_{13})^t$ in 3d, with $\epsilon_{ij}(\boldsymbol{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$. Then the Hooke's law writes

$$\underline{\boldsymbol{\sigma}} = \mathbb{C}\underline{\boldsymbol{\epsilon}},$$

where $\underline{\boldsymbol{\sigma}}$ is the elastic stress tensor and $\mathbb{C}$ the elasticity tensor.

The material is supposed to be isotropic. Thus the elasticity tensor $\mathbb{C}$ is only defined by the Lamé parameters λ and μ, which satisfy $\lambda + \mu > 0$. We also set $\gamma = 2\,\mu + \lambda$. For $d = 2$ or $d = 3$, $\mathbb{C}$ is given by

$$\mathbb{C} = \begin{pmatrix} \lambda\mathbb{1}_2 + 2\mu\mathbb{I}_2 & 0 \\ 0 & \mu \end{pmatrix}_{3\times 3} \quad \text{or} \quad \mathbb{C} = \begin{pmatrix} \lambda\mathbb{1}_3 + 2\mu\mathbb{I}_3 & 0 \\ 0 & \mu\mathbb{I}_3 \end{pmatrix}_{6\times 6},$$

respectively, where $\mathbb{1}_d$ is a d-by-d matrix of ones, and $\mathbb{I}_d$ the d-by-d identity matrix.

For dimension $d = 2$ or $d = 3$, we have:

$$\boldsymbol{\sigma}_{\alpha\beta}(\boldsymbol{u}) \quad = \quad 2\,\mu\,\boldsymbol{\epsilon}_{\alpha\beta}(\boldsymbol{u}) + \lambda\,\mathrm{tr}(\boldsymbol{\epsilon}(\boldsymbol{u}))\delta_{\alpha\beta} \quad \forall \alpha, \beta \in [\![1, d]\!]$$

The problem to solve is the following

Usual vector BVP 2 : Elasticity problem

Find $\boldsymbol{u} = \mathrm{H}^2(\Omega)^d$ such that

$$-\operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u})) = \boldsymbol{f}, \quad \text{in } \Omega \subset \mathbb{R}^d, \tag{4.1}$$

$$\boldsymbol{\sigma}(\boldsymbol{u}).\boldsymbol{n} = \boldsymbol{0} \quad \text{on } \Gamma^R, \tag{4.2}$$

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } \Gamma^D. \tag{4.3}$$

Now, with the following lemma, we obtain that this problem can be rewritten as the vector BVP defined by (1.14) to (1.16).

Lemme 4.1

Let $\mathcal{H}$ be the d-by-d matrix of the second order linear differential operators defined in (1.10) where $\mathcal{H}_{\alpha,\beta} = \mathcal{L}_{\mathbb{A}^{\alpha,\beta},\mathbf{0},\mathbf{0},0}$, $\forall(\alpha,\beta) \in [\![1,d]\!]^2$, with

$$(\mathbb{A}^{\alpha,\beta})_{k,l} = \mu\delta_{\alpha\beta}\delta_{kl} + \mu\delta_{k\beta}\delta_{l\alpha} + \lambda\delta_{k\alpha}\delta_{l\beta},\ \forall(k,l) \in [\![1,d]\!]^2. \tag{4.4}$$

then

$$\mathcal{H}(\boldsymbol{u}) = -\operatorname{div}\boldsymbol{\sigma}(\boldsymbol{u}) \tag{4.5}$$

and, $\forall\alpha \in [\![1,d]\!]$,

$$\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}_\alpha}} = (\boldsymbol{\sigma}(\boldsymbol{u}).\boldsymbol{n})_\alpha. \tag{4.6}$$

The proof is given in appendix ??. So we obtain

***Vector* BVP 3 : Elasticity problem with $\mathcal{H}$ operator in dimension $d = 2$ or $d = 3$**

Let $\mathcal{H}$ be the d-by-d matrix of the second order linear differential operators defined in (1.10) where $\forall(\alpha,\beta) \in [\![1,d]\!]^2$, $\mathcal{H}_{\alpha,\beta} = \mathcal{L}_{\mathbb{A}^{\alpha,\beta},\mathbf{0},\mathbf{0},0}$, with

- for $d = 2$,
$$\mathbb{A}^{1,1} = \begin{pmatrix}\gamma & 0\\ 0 & \mu\end{pmatrix},\ \mathbb{A}^{1,2} = \begin{pmatrix}0 & \lambda\\ \mu & 0\end{pmatrix},\ \mathbb{A}^{2,1} = \begin{pmatrix}0 & \mu\\ \lambda & 0\end{pmatrix},\ \mathbb{A}^{2,2} = \begin{pmatrix}\mu & 0\\ 0 & \gamma\end{pmatrix}$$
- for $d = 3$,
$$\mathbb{A}^{1,1} = \begin{pmatrix}\gamma & 0 & 0\\ 0 & \mu & 0\\ 0 & 0 & \mu\end{pmatrix},\quad \mathbb{A}^{1,2} = \begin{pmatrix}0 & \lambda & 0\\ \mu & 0 & 0\\ 0 & 0 & 0\end{pmatrix},\quad \mathbb{A}^{1,3} = \begin{pmatrix}0 & 0 & \lambda\\ 0 & 0 & 0\\ \mu & 0 & 0\end{pmatrix}$$
$$\mathbb{A}^{2,1} = \begin{pmatrix}0 & \mu & 0\\ \lambda & 0 & 0\\ 0 & 0 & 0\end{pmatrix},\quad \mathbb{A}^{2,2} = \begin{pmatrix}\mu & 0 & 0\\ 0 & \gamma & 0\\ 0 & 0 & \mu\end{pmatrix},\quad \mathbb{A}^{2,3} = \begin{pmatrix}0 & 0 & 0\\ 0 & 0 & \lambda\\ 0 & \mu & 0\end{pmatrix},$$
$$\mathbb{A}^{3,1} = \begin{pmatrix}0 & 0 & \mu\\ 0 & 0 & 0\\ \lambda & 0 & 0\end{pmatrix},\quad \mathbb{A}^{3,2} = \begin{pmatrix}0 & 0 & 0\\ 0 & 0 & \mu\\ 0 & \lambda & 0\end{pmatrix},\quad \mathbb{A}^{3,3} = \begin{pmatrix}\mu & 0 & 0\\ 0 & \mu & 0\\ 0 & 0 & \gamma\end{pmatrix}.$$

The elasticity problem (4.1) to (4.3) can be rewritten as :
Find $\boldsymbol{u} = (\boldsymbol{u}_1,\ldots,\boldsymbol{u}_d) \in (\mathrm{H}^2(\Omega))^d$ such that

$$\mathcal{H}(\boldsymbol{u}) = \boldsymbol{f}, \quad \text{in } \Omega, \tag{4.7}$$

$$\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}_\alpha}} = 0, \quad \text{on } \Gamma^R_\alpha = \Gamma^R,\ \forall\alpha \in [\![1,d]\!] \tag{4.8}$$

$$\boldsymbol{u}_\alpha = 0, \quad \text{on } \Gamma^D_\alpha = \Gamma^D,\ \forall\alpha \in [\![1,d]\!]. \tag{4.9}$$

4.1.2 2D example

For example, in 2d, we want to solve the elasticity problem (4.1) to (4.3) where Ω and its boundaries are given in Figure 4.1.

The material's properties are given by Young's modulus E and Poisson's coefficient ν. As we use plane strain hypothesis, Lame's coefficients verify

$$\mu = \frac{E}{2\,(1+\nu)},\quad \lambda = \frac{E\,\nu}{(1+\nu)\,(1-2\,\nu)},\quad \gamma = 2\mu + \lambda$$

The material is rubber so that $E = 21\,.10^5$Pa and $\nu = 0.45$. We also have $\boldsymbol{f} = \boldsymbol{x} \mapsto (0,-1)^t$ and we choose $\Gamma^R = \Gamma^1 \cup \Gamma^2 \cup \Gamma^3$, $\Gamma^D = \Gamma^4$.

We give in Listing 4.1 the corresponding Octave codes.

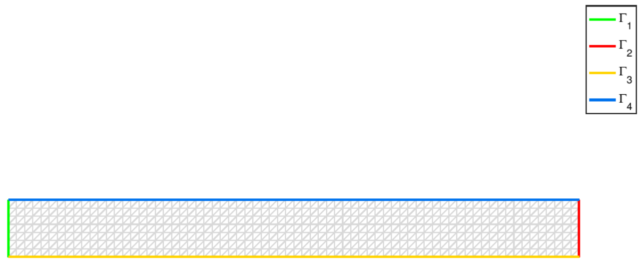


Figure 4.1: Domain and boundaries

Listing 4.1: **2D elasticity, Matlab code**

```
fprintf('1. Building the mesh using HyperCube function\n');
%Hop=Hoperator.StiffElas(dim,lam,mu);
gamma=lambda+2*mu;
Hop=Hoperator(dim,dim,dim);
Hop.set(1,1,Loperator(dim,dim,{gamma,[];[],mu},[],[],[]));
Hop.set(1,2,Loperator(dim,dim,{[],lambda;mu,[]},[],[],[]));
Hop.set(2,1,Loperator(dim,dim,{[],mu;lambda,[]},[],[],[]));
Hop.set(2,2,Loperator(dim,dim,{mu,[];[],gamma},[],[],[]));
pde=PDEelt(Hop,{0,-1});
bvp=BVP(Th,pde);
fprintf('2.b Solving 2D elasticity BVP\n')
```

One can also use the Octave function HOPERATOR.STIFFELAS to build the elasticity operator :

```
Hop=Hoperator.StiffElas(dim,lambda,mu);
```

For a given mesh, its displacement scaled by a factor 50 is shown on Figure 4.2

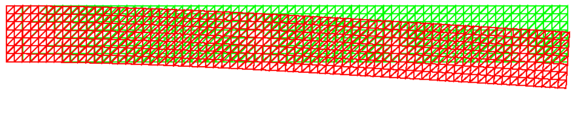


Figure 4.2: Mesh displacement scaled by a factor 50 for the 2D elasticity problem

4.1.3 3D example

Let $\Omega = [0,5] \times [0,1] \times [0,1] \subset \mathbb{R}^3$. The boundary of Ω is made of six faces and each one has a unique label : 1 to 6 respectively for faces $x_1 = 0$, $x_1 = 5$, $x_2 = 0$, $x_2 = 1$, $x_3 = 0$ and $x_3 = 1$. We represent them in Figure 4.3.

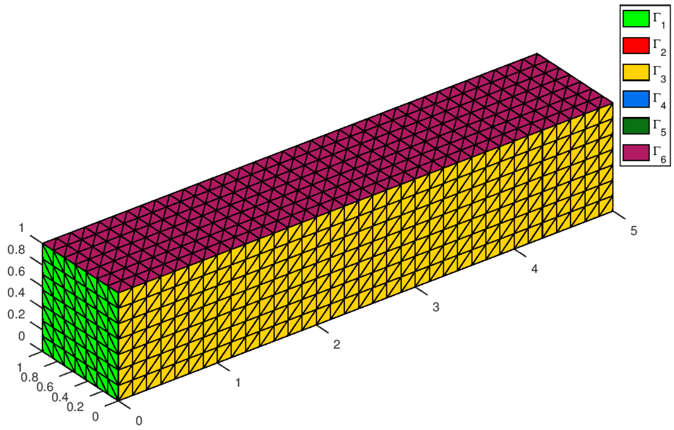


Figure 4.3: Domain and boundaries

We want to solve the elasticity problem (4.1) to (4.3) with $\Gamma^D = \Gamma_1$, $\Gamma^N = \bigcup_{i=2}^{6} \Gamma_i$ and $\boldsymbol{f} = \boldsymbol{x} \mapsto (0,0,-1)^t$.

We give in Listing 4.2 the corresponding Octave code using function HOPERATOR.STIFFELAS .

Listing 4.2: **3D elasticity, Octave code**

```
fprintf('1. Building the mesh using HyperCube function\n');
fprintf('2.a Setting 3D elasticity BVP\n');
Hop=Hoperator();
Hop.opStiffElas(dim,lambda,mu);
pde=PDEelt(Hop,{0,0,−1});
bvp.setDirichlet(1,0.,1:3);
```

The displacement scaled by a factor 2000 for a given mesh is shown on Figure 4.4.

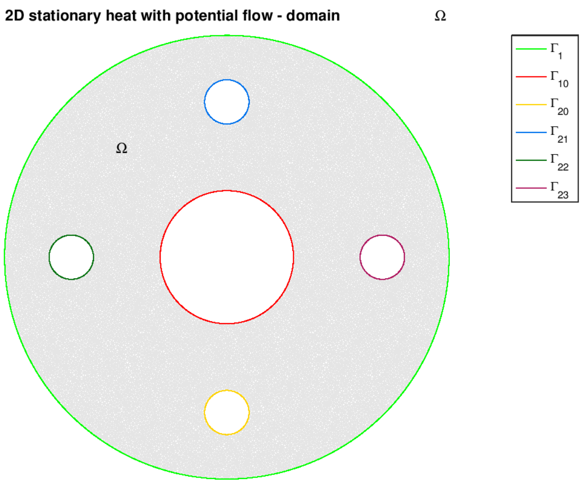


Figure 4.5: Domain and boundaries

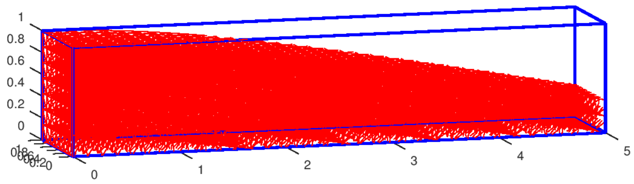


Figure 4.4: Result for the 3D elasticity problem

4.2 Stationary heat with potential flow in 2D

Let Γ_1 be the unit circle, Γ_{10} be the circle with center point $(0,0)$ and radius 0.3. Let Γ_{20}, Γ_{21}, Γ_{22} and Γ_{23} be the circles with radius 0.1 and respectively with center point $(0,-0.7)$, $(0,0.7)$, $(-0.7,0)$ and $(0.7,0)$. The domain $\Omega \subset \mathbb{R}^2$ is defined as the inner of Γ_1 and the outer of all other circles (see Figure 4.5).

The 2D problem to solve is the following

Usual BVP 6 : 2D problem : stationary heat with potential flow

Find $u \in H^2(\Omega)$ such that

$$-\operatorname{div}(\alpha \nabla u) + \langle \boldsymbol{V}, \nabla u\rangle + \beta u = 0 \ \text{ in } \Omega \subset \mathbb{R}^2, \tag{4.10}$$

$$u = 20 * \boldsymbol{x}_2 \ \text{ on } \Gamma_{21}, \tag{4.11}$$

$$u = 0 \ \text{ on } \Gamma_{22} \cup \Gamma_{23}, \tag{4.12}$$

$$\frac{\partial u}{\partial n} = 0 \ \text{ on } \Gamma_1 \cup \Gamma_{10} \cup \Gamma_{20} \tag{4.13}$$

where Ω and its boundaries are given in Figure 4.5. This problem is well posed if $\alpha(\boldsymbol{x}) > 0$ and $\beta(\boldsymbol{x}) \geqslant 0$.
We choose α and β in Ω as :

$$\alpha(\boldsymbol{x}) = 0.1 + \boldsymbol{x}_2^2,$$
$$\beta(\boldsymbol{x}) = 0.01$$

The potential flow is the velocity field $\boldsymbol{V} = \nabla \phi$ where the scalar function ϕ is the velocity potential solution of the 2D BVP (4.14)-(4.17)

Usual BVP 7 : 2D velocity potential BVP

Find $\phi \in H^2(\Omega)$ such that

$$-\Delta\phi = 0 \ \text{ in } \Omega, \tag{4.14}$$

$$\phi = -20 \ \text{ on } \Gamma_{21}, \tag{4.15}$$

$$\phi = 20 \ \text{ on } \Gamma_{20}, \tag{4.16}$$

$$\frac{\partial \phi}{\partial n} = 0 \ \text{ on } \Gamma_1 \cup \Gamma_{23} \cup \Gamma_{22} \tag{4.17}$$

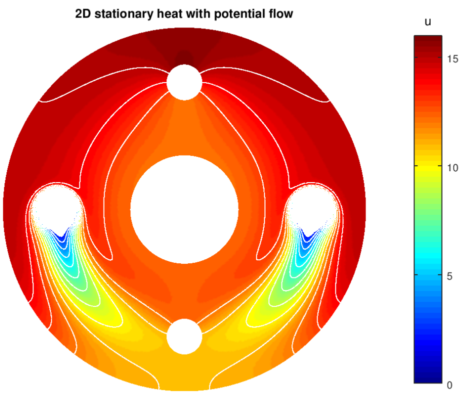
Then the potential flow $\boldsymbol{V}$ is *solution* of (4.18)

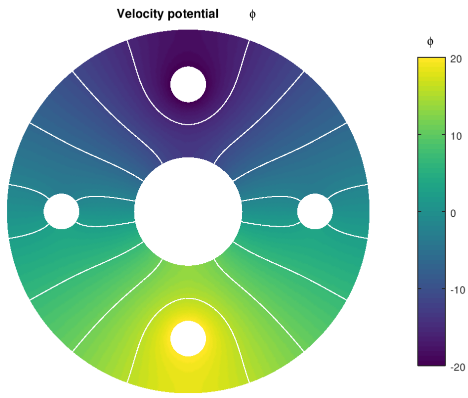
Usual vector BVP 3 : 2D potential flow

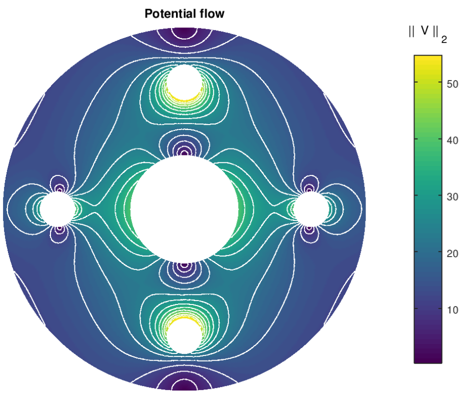
Find $\boldsymbol{V} = (\boldsymbol{V}_1, \boldsymbol{V}_2) \in H^1(\Omega) \times H^1(\Omega)$ such that

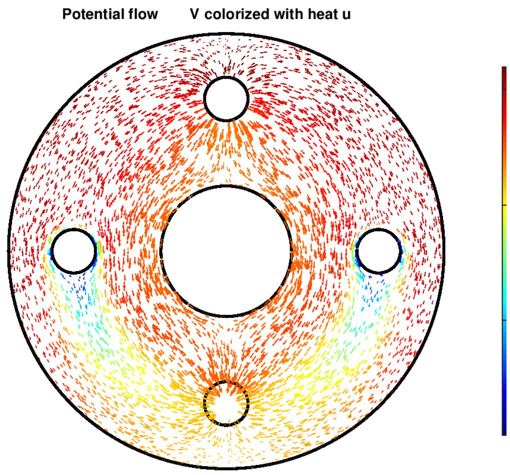
$$\boldsymbol{V} = \nabla \phi \ \text{ in } \Omega, \tag{4.18}$$

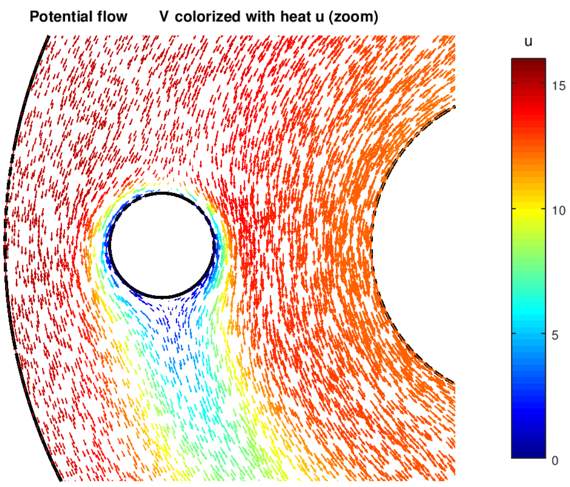
For a given mesh, the numerical result for heat u is represented in Figure **??**, velocity potential ϕ and potential flow $\boldsymbol{V}$ are shown on Figure **??**.











Now we will present two manners of solving these problems using FC-VFEM$\mathbb{P}_1$ codes.

4.2.1 Method 1 : split in three parts

The 2D potential velocity problem (4.14)-(4.17) can be equivalently expressed as the scalar BVP (1.2)-(1.4) :

Scalar BVP 9 : 2D potential velocity

Find $\phi \in \mathrm{H}^2(\Omega)$ such that

$$\begin{aligned} \mathcal{L}(\phi) &= f && \text{in } \Omega, \\ \phi &= g^D && \text{on } \Gamma^D, \\ \frac{\partial \phi}{\partial n_{\mathcal{L}}} + a^R \phi &= g^R && \text{on } \Gamma^R. \end{aligned}$$

where

- $\mathcal{L} := \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}$, and then the conormal derivative of ϕ is given by

$$\frac{\partial \phi}{\partial n_{\mathcal{L}}} := \langle \mathbb{A} \nabla \phi, \boldsymbol{n} \rangle - \langle \boldsymbol{b}\phi, \boldsymbol{n} \rangle = \frac{\partial \phi}{\partial n}.$$

- $f(\boldsymbol{x}) := 0$
- $\Gamma^D = \Gamma_{20} \cup \Gamma_{21}$
- $\Gamma^R = \Gamma_1 \cup \Gamma_{23} \cup \Gamma_{22}$
- $g^D := 20$ on Γ_{20}, and $g^D := -20$ on Γ_{21}
- $g^R = a^R := 0$ on Γ^R. (Neumann boundary condition)

The code using the toolbox for solving (4.14)-(4.17) is given in Listing 4.6.

Listing 4.3: **Stationary heat with potential flow in 2D, Octave code (method 1)**

```
d=2;
Lop=Loperator(d,d,{1,[];[],1},[],[],[]);
bvpPotential=BVP(Th,PDEelt(Lop));
bvpPotential.setDirichlet(20,20);
bvpPotential.setDirichlet(21,-20);
phi=bvpPotential.solve();
```

Now to compute $\boldsymbol{V}$, we can write the potential flow problem (4.18) with $\mathcal{H}$-operators as

$$\boldsymbol{V} = \begin{pmatrix} \boldsymbol{V}_1 \\ \boldsymbol{V}_2 \end{pmatrix} = \mathcal{B} \begin{pmatrix} \phi \\ \phi \end{pmatrix}$$

where

$$\mathcal{B} = \begin{pmatrix} \mathcal{L}_{\mathbb{O}_2,\mathbf{0}_2,(1,0)^t,1} & 0 \\ 0 & \mathcal{L}_{\mathbb{O}_2,\mathbf{0}_2,(0,1)^t,0} \end{pmatrix}$$

The code using the toolbox for solving this problem is given in Listing 4.6.

Listing 4.4: **Stationary heat with potential flow in 2D, Octave code (method 1)**

```
Hop=Hoperator(Th.dim,d,d);
Hop.H{1,1}=Loperator(d,d,[],[],{1,0},[]);
Hop.H{2,2}=Loperator(d,d,[],[],{0,1},[]);
V=Hop.apply(Th,{phi,phi});
```

Obviously, one can compute separately $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$.

Finally, the stationary heat BVP (4.10)-(4.13) can be equivalently expressed as the scalar BVP (1.2)-(1.4) :

Usual BVP 8 : 2D stationary heat

Find $u \in \mathrm{H}^2(\Omega)$ such that

$$\begin{aligned} \mathcal{L}(u) &= f && \text{in } \Omega, \\ u &= g^D && \text{on } \Gamma^D, \\ \frac{\partial u}{\partial n_{\mathcal{L}}} + a^R u &= g^R && \text{on } \Gamma^R. \end{aligned}$$

where

- $\mathcal{L} := \mathcal{L}_{\begin{pmatrix}\alpha & 0\\ 0 & \alpha\end{pmatrix},\mathbf{0},\boldsymbol{V},\beta}$, and then the conormal derivative of u is given by

$$\frac{\partial u}{\partial n_{\mathcal{L}}} := \langle \mathbb{A}\boldsymbol{\nabla}\, u, \boldsymbol{n}\rangle - \langle \boldsymbol{b}u, \boldsymbol{n}\rangle = \alpha\frac{\partial u}{\partial n}.$$

- $f := 0$
- $\Gamma^D = \Gamma_{21} \cup \Gamma_{22} \cup \Gamma_{23}$
- $\Gamma^R = \Gamma_1 \cup \Gamma_{10} \cup \Gamma_{20}$
- $g^D(x,y) := 20y$ on Γ_{21}, and $g^D := 0$ on $\Gamma_{22} \cup \Gamma_{23}$
- $g^R := 0$ and $a^R := 0$ on Γ^R

The code using the package FC-VFEM$\mathbb{P}_1$ for solving (4.10)-(4.13) is given in Listing 4.6.

Listing 4.5: **Stationary heat with potential flow in 2D, Octavecode (method 1)**

```
Lop=Loperator(d,d,{af,[];[],af},[],V,b);
bvpHeat=BVP(Th,PDEelt(Lop));
bvpHeat.setDirichlet(21,gD);
bvpHeat.setDirichlet(22, 0);
bvpHeat.setDirichlet(23, 0);
u=bvpHeat.solve();
```

4.2.2 Method 2 : have fun with $\mathcal{H}$-operators

We can merged velocity potential BVP (4.14)-(4.17) and potential flow to obtain the new BVP

Usual vector BVP 4 : Velocity potential and potential flow in 2D

Find $\phi \in \mathrm{H}^2(\Omega)$ and $\boldsymbol{V} = (\boldsymbol{V}_1, \boldsymbol{V}_2) \in \mathrm{H}^1(\Omega) \times \mathrm{H}^1(\Omega)$ such that

$$-\left(\frac{\partial \boldsymbol{V}_1}{\partial x} + \frac{\partial \boldsymbol{V}_2}{\partial y}\right) = 0 \text{ in } \Omega, \tag{4.19}$$

$$\boldsymbol{V}_1 - \frac{\partial \phi}{\partial x} = 0 \text{ in } \Omega, \tag{4.20}$$

$$\boldsymbol{V}_2 - \frac{\partial \phi}{\partial y} = 0 \text{ in } \Omega, \tag{4.21}$$

$$\phi = -20 \text{ on } \Gamma_{21}, \tag{4.22}$$

$$\phi = 20 \text{ on } \Gamma_{20}, \tag{4.23}$$

$$\frac{\partial \phi}{\partial n} = 0 \text{ on } \Gamma_1 \cup \Gamma_{23} \cup \Gamma_{22} \tag{4.24}$$

We can also replace (4.19) by $-\Delta\phi = 0$.

Let $\boldsymbol{w} = \begin{pmatrix}\phi\\ \boldsymbol{V}_1\\ \boldsymbol{V}_2\end{pmatrix}$, the previous problem (4.19)-(4.24) can be equivalently expressed as the vector BVP (1.14)-(1.16) :

Vector BVP 4 : Velocity potential and potential flow in 2D

Find $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3) \in (\mathrm{H}^2(\Omega))^3$ such that

$$\mathcal{H}(\boldsymbol{w}) = \boldsymbol{f} \quad \text{in } \Omega, \tag{4.25}$$

$$\boldsymbol{w}_\alpha = g^D_\alpha \quad \text{on } \Gamma^D_\alpha,\ \forall \alpha \in [\![1,3]\!], \tag{4.26}$$

$$\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{H}_\alpha}} + a^R_\alpha \boldsymbol{w}_\alpha = g^R_\alpha \quad \text{on } \Gamma^R_\alpha,\ \forall \alpha \in [\![1,3]\!], \tag{4.27}$$

where $\Gamma^R_\alpha = \Gamma^D_\alpha = \varnothing$ for all $\alpha \in \{2,3\}$ (no boundary conditions on $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$) and

- $\mathcal{H}$ is the 3-by-3 operator defined by

$$\mathcal{H} = \begin{pmatrix} 0 & \mathcal{L}_{\mathbb{O},-\boldsymbol{e}_1,\boldsymbol{0},0} & \mathcal{L}_{\mathbb{O},-\boldsymbol{e}_2,\boldsymbol{0},0} \\ \mathcal{L}_{\mathbb{O},\boldsymbol{0},-\boldsymbol{e}_1,0} & \mathcal{L}_{\mathbb{O},\boldsymbol{0},\boldsymbol{0},1} & 0 \\ \mathcal{L}_{\mathbb{O},\boldsymbol{0},-\boldsymbol{e}_2,0} & 0 & \mathcal{L}_{\mathbb{O},\boldsymbol{0},\boldsymbol{0},1} \end{pmatrix}$$

its conormal derivative are given by

$$\frac{\partial \boldsymbol{w}_1}{\partial n_{\mathcal{H}_{1,1}}} = 0, \qquad \frac{\partial \boldsymbol{w}_2}{\partial n_{\mathcal{H}_{1,2}}} = \boldsymbol{w}_2 \boldsymbol{n}_1, \qquad \frac{\partial \boldsymbol{w}_3}{\partial n_{\mathcal{H}_{1,3}}} = \boldsymbol{w}_3 \boldsymbol{n}_2,$$
$$\frac{\partial \boldsymbol{w}_1}{\partial n_{\mathcal{H}_{2,1}}} = 0, \qquad \frac{\partial \boldsymbol{w}_2}{\partial n_{\mathcal{H}_{2,2}}} = 0, \qquad \frac{\partial \boldsymbol{w}_3}{\partial n_{\mathcal{H}_{2,3}}} = 0$$
$$\frac{\partial \boldsymbol{w}_1}{\partial n_{\mathcal{H}_{3,1}}} = 0, \qquad \frac{\partial \boldsymbol{w}_2}{\partial n_{\mathcal{H}_{3,2}}} = 0, \qquad \frac{\partial \boldsymbol{w}_3}{\partial n_{\mathcal{H}_{3,3}}} = 0.$$

So we obtain

$$\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{H}_1}} \stackrel{\text{def}}{=} \sum_{\alpha=1}^{3} \frac{\partial \boldsymbol{w}_\alpha}{\partial n_{\mathcal{H}_{1,\alpha}}} = \langle \boldsymbol{V}, \boldsymbol{n} \rangle = \frac{\partial \phi}{\partial \boldsymbol{n}}, \tag{4.28}$$

and

$$\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{H}_2}} = \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{H}_3}} := 0. \tag{4.29}$$

From (4.29), we cannot impose boundary conditions on components 2 and 3.

- $\boldsymbol{f} := \boldsymbol{0}$
- $\Gamma_1^D = \Gamma_{20} \cup \Gamma_{21}$ and $\Gamma_1^R = \Gamma_1 \cup \Gamma_{10} \cup \Gamma_{22} \cup \Gamma_{23}$
- $g_1^D := 20$ on Γ_{20}, and $g_1^D := -20$ on Γ_{21}
- $g_1^R = a_1^R := 0$ on Γ_1^R

The solution of this vector BVP is obtain by using the Octave code is given by Listing 4.6.

Listing 4.6: **Stationary heat with potential flow in 2D, Octave code (method 1)**

```
d=2;
Hop=Hoperator(d,d,3);
Hop.set(1,2,Loperator(d,d,[],{−1,0},[],[]));
Hop.set(1,3,Loperator(d,d,[],{0,−1},[],[]));
Hop.set(2,1,Loperator(d,d,[],[],{−1,0},[]));
Hop.set(2,2,Loperator(d,d,[],[],[],1));
Hop.set(3,1,Loperator(d,d,[],[],{0,−1},[]));
Hop.set(3,3,Loperator(d,d,[],[],[],1));
bvpFlow=BVP(Th,PDEelt(Hop));
bvpFlow.setDirichlet(20,20,1);
bvpFlow.setDirichlet(21,−20,1);
U=bvpFlow.solve('split',true);
```

4.3 Stationary heat with potential flow in 3D

Let $\Omega \subset \mathbb{R}^3$ be the cylinder given in Figure 4.6.

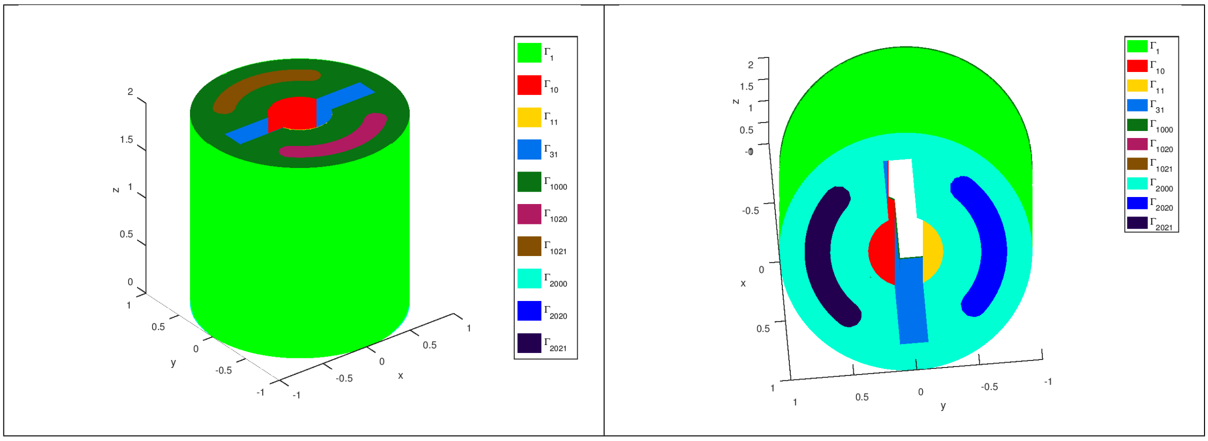


Figure 4.6: Stationary heat with potential flow : 3d mesh

The bottom and top faces of the cylinder are respectively $\Gamma_{1000} \cup \Gamma_{1020} \cup \Gamma_{1021}$ and $\Gamma_{2000} \cup \Gamma_{2020} \cup \Gamma_{2021}$. The hole surface is $\Gamma_{10} \cup \Gamma_{11} \cup \Gamma_{31}$ where $\Gamma_{10} \cup \Gamma_{11}$ is the cylinder part and Γ_{31} the plane part.

The 3D problem to solve is the following

Usual BVP 9 : 3D stationary heat with potential flow

Find $u \in \mathrm{H}^2(\Omega)$ such that

$$
\begin{aligned}
-\operatorname{div}(\alpha \nabla u) + \langle \boldsymbol{V}, \nabla u \rangle + \beta u &= 0 \quad \text{in } \Omega \subset \mathbb{R}^3, &(4.30)\\
u &= 30 \quad \text{on } \Gamma_{1020} \cup \Gamma_{2020}, &(4.31)\\
u &= 10\delta_{|z-1|>0.5} \quad \text{on } \Gamma_{10}, &(4.32)\\
\frac{\partial u}{\partial n} &= 0 \quad \text{otherwise} &(4.33)
\end{aligned}
$$

where Ω and its boundaries are given in Figure 4.6. This problem is well posed if $\alpha(\boldsymbol{x}) > 0$ and $\beta(\boldsymbol{x}) \geqslant 0$.
We choose α and β in Ω as :

$$
\begin{aligned}
\alpha(\boldsymbol{x}) &= 1 + (x_3 - 1)^2; ,\\
\beta(\boldsymbol{x}) &= 0.01
\end{aligned}
$$

The potential flow is the velocity field $\boldsymbol{V} = \nabla \phi$ where the scalar function ϕ is the velocity potential solution of the 3D BVP (4.34)-(4.37)

Usual BVP 10 : 3D velocity potential

Find $\phi \in \mathrm{H}^1(\Omega)$ such that

$$
\begin{aligned}
-\Delta \phi &= 0 \quad \text{in } \Omega, &(4.34)\\
\phi &= 1 \quad \text{on } \Gamma_{1021} \cup \Gamma_{2021}, &(4.35)\\
\phi &= -1 \quad \text{on } \Gamma_{1020} \cup \Gamma_{2020}, &(4.36)\\
\frac{\partial \phi}{\partial n} &= 0 \quad \text{otherwise} &(4.37)
\end{aligned}
$$

Then the potential flow $\boldsymbol{V}$ is *solution* of (4.38)

Usual vector BVP 5 : 3D potential flow

Find $\boldsymbol{V} = (\boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_3) \in \mathrm{H}^1(\Omega) \times \mathrm{H}^1(\Omega)$ such that

$$
\boldsymbol{V} = \nabla \phi \quad \text{in } \Omega, \qquad (4.38)
$$

For a given mesh, the numerical result for heat u is represented in Figure 4.7, velocity potential ϕ and potential flow $\boldsymbol{V}$ are shown in Figure 4.8.

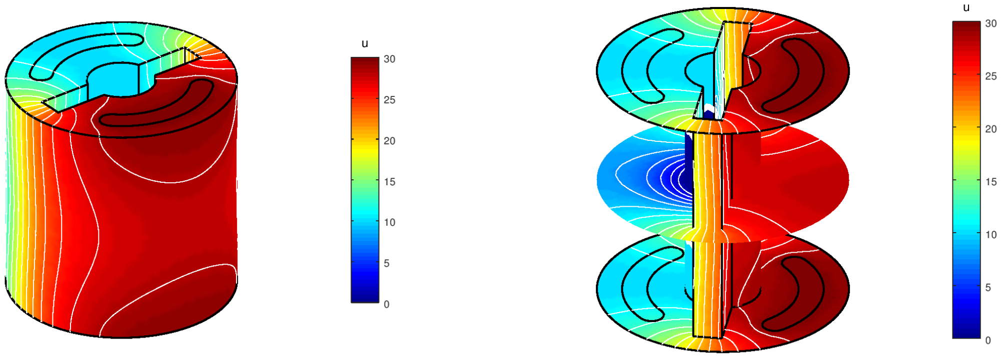


Figure 4.7: Heat solution u

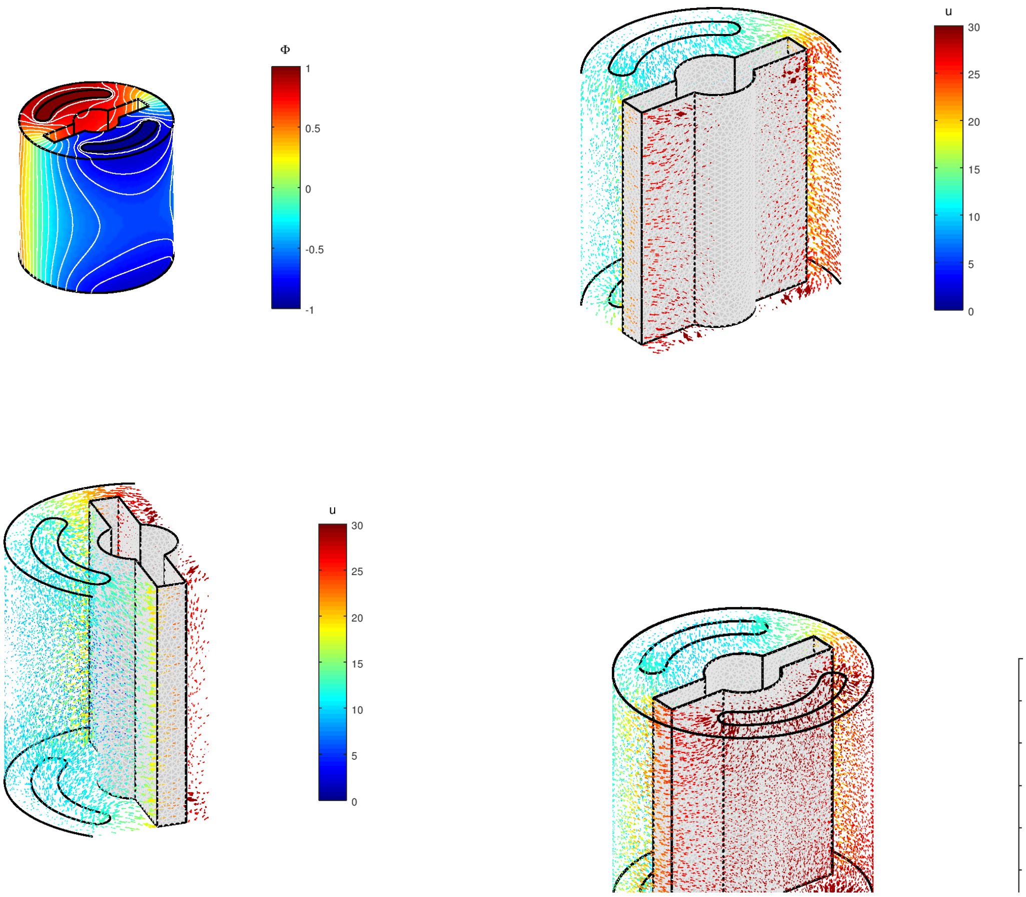


Figure 4.8: Velocity potential Φ (bottom) and velocity field $\boldsymbol{V} = \boldsymbol{\nabla}\,\Phi$ (upper)

Now we will present two manners of solving these problems using FC-VFEM$\mathbb{P}_1$ codes.

4.3.1 Method 1 : split in three parts

The 3D potential velocity problem (4.34)-(4.37) can be equivalently expressed as the scalar BVP (1.2)-(1.4) :

Scalar BVP 10 : 3D potential velocity

Find $\phi \in H^1(\Omega)$ such that

$$\begin{aligned}\mathcal{L}(\phi) &= f && \text{in } \Omega,\\ \phi &= g^D && \text{on } \Gamma^D,\\ \frac{\partial \phi}{\partial n_{\mathcal{L}}} + a^R\phi &= g^R && \text{on } \Gamma^R.\end{aligned}$$

where

- $\mathcal{L} := \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}$, and then the conormal derivative of ϕ is given by

$$\frac{\partial \phi}{\partial n_{\mathcal{L}}} := \langle \mathbb{A}\boldsymbol{\nabla}\phi, \boldsymbol{n}\rangle - \langle \boldsymbol{b}\phi, \boldsymbol{n}\rangle = \frac{\partial \phi}{\partial n}.$$

- $f(\boldsymbol{x}) := 0$
- $\Gamma^D = \Gamma_{1020} \cup \Gamma_{1021} \cup \Gamma_{2020} \cup \Gamma_{2021}$
- $\Gamma^R = \Gamma_1 \cup \Gamma_{10} \cup \Gamma_{11} \cup \Gamma_{31} \cup \Gamma_{1000} \cup \Gamma_{2000}$
- $g^D := 1$ on $\Gamma_{1021} \cup \Gamma_{2021}$, and $g^D := -1$ on $\Gamma_{1020} \cup \Gamma_{2020}$
- $g^R = a^R := 0$ on Γ^R. (Neumann boundary condition)

The code using the package for solving (4.34)-(4.37) is given in Listing 4.7

Listing 4.7: **Stationary heat with potential flow in 3D, Octave code (method 1)**

```
d=3;dim=3;
Lop=Loperator(dim,d, {1,[],[];[],1,[];[],[],1},[],[],[]);
bvpFlow=BVP(Th,PDEelt(Lop));
bvpFlow.setDirichlet(1021,1.);
bvpFlow.setDirichlet(2021,1.);
bvpFlow.setDirichlet(1020,−1.);
bvpFlow.setDirichlet(2020,−1.);
Phi=bvpFlow.solve();
```

Now to compute $\boldsymbol{V}$, we can write the potential flow problem (4.38)

- with $\mathcal{H}$-operators as

$$\boldsymbol{V} = \begin{pmatrix}\boldsymbol{V}_1\\ \boldsymbol{V}_2\\ \boldsymbol{V}_2\end{pmatrix} = \mathcal{B}\begin{pmatrix}\phi\\ \phi\\ \phi\end{pmatrix}$$

where

$$\mathcal{B} = \begin{pmatrix}\mathcal{L}_{\mathbb{O}_3,\mathbf{0}_3,(1,0,0)^t,1} & 0 & 0\\ 0 & \mathcal{L}_{\mathbb{O}_3,\mathbf{0}_3,(0,1,0)^t,0} & 0\\ 0 & 0 & \mathcal{L}_{\mathbb{O}_3,\mathbf{0}_3,(0,0,1)^t,0}\end{pmatrix}$$

- with $\mathcal{L}$-operators as

$$\boldsymbol{V} = \begin{pmatrix}\boldsymbol{V}_1\\ \boldsymbol{V}_2\\ \boldsymbol{V}_2\end{pmatrix} = \boldsymbol{\nabla}\phi = \begin{pmatrix}\mathcal{L}_{\mathbb{O}_3,\mathbf{0}_3,(1,0,0)^t,0}(\phi)\\ \mathcal{L}_{\mathbb{O}_3,\mathbf{0}_3,(0,1,0)^t,0}(\phi)\\ \mathcal{L}_{\mathbb{O}_3,\mathbf{0}_3,(0,0,1)^t,0}(\phi)\end{pmatrix}$$

The code using FC-VFEM$\mathbb{P}_1$ package for solving this problem with $\mathcal{L}$-operators is given in Listing 4.8.

Listing 4.8: **Stationary heat with potential flow in 3D, Octave code (method 1)**

```
Lop=Loperator(dim,d,[],[],{1,0,0},[]);
V{1}=Lop.apply(Th,Phi);
Lop=Loperator(dim,d,[],[],{0,1,0},[]);
V{2}=Lop.apply(Th,Phi);
Lop=Loperator(dim,d,[],[],{0,0,1},[]);
V{3}=Lop.apply(Th,Phi);
```

Finally, the stationary heat BVP (4.30)-(**??**) can be equivalently expressed as the scalar BVP (1.2)-(1.4) :

Scalar BVP 11 : 3D stationary heat

Find $u \in \mathrm{H}^1(\Omega)$ such that

$$\begin{aligned} \mathcal{L}(u) &= f && \text{in } \Omega, \\ u &= g^D && \text{on } \Gamma^D, \\ \frac{\partial u}{\partial n_{\mathcal{L}}} + a^R u &= g^R && \text{on } \Gamma^R. \end{aligned}$$

where

- $\mathcal{L} := \mathcal{L}_{\begin{pmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{pmatrix}, \mathbf{0}, \boldsymbol{V}, \beta}$, and then the conormal derivative of u is given by

$$\frac{\partial u}{\partial n_{\mathcal{L}}} := \langle \mathbb{A} \boldsymbol{\nabla} u, \boldsymbol{n} \rangle - \langle \boldsymbol{b} u, \boldsymbol{n} \rangle = \alpha \frac{\partial u}{\partial n}.$$

- $f := 0$
- $\Gamma^D = \Gamma_{1020} \cup \Gamma_{2020} \cup \Gamma_{10}$
- $\Gamma^R = \Gamma_1 \cup \Gamma_{11} \cup \Gamma_{31} \cup \Gamma_{1000} \cup \Gamma_{1021} \cup \Gamma_{2000} \cup \Gamma_{2021}$
- $g^D(x,y,z) := 30$ on $\Gamma_{1020} \cup \Gamma_{2020}$, and $g^D(x,y,z) := 10(|z-1| > 0.5)$ on Γ_{10}
- $g^R := 0$ and $a^R := 0$ on Γ^R

The code using the package for solving (4.30)-(**??**) is given in Figure 4.9.

Listing 4.9: **Stationary heat with potential flow in 3D, Octave code (method 1)**

```
af=@(x,y,z) 1+(z-1).^2;
Lop=Loperator(dim,d, {af,[],[];[],af,[];[],[],af},[], {V{1},V{2},V{3}},0.01);
bvpHeat=BVP(Th,PDEelt(Lop));
bvpHeat.setDirichlet(1020,30.);
bvpHeat.setDirichlet(2020,30.);
bvpHeat.setDirichlet(10, @(x,y,z) 10*(abs(z-1)>0.5));
U=bvpHeat.solve();
```

4.3.2 Method 2 : have fun with $\mathcal{H}$-operators

To solve problem (4.30)-(4.33), we need to compute the velocity field $\boldsymbol{V}$. For that we can rewrite the potential flow problem (4.34)-(4.37), by introducing $\boldsymbol{V} = (\boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_3)$ as unknowns :

Usual vector BVP 6 : Velocity potential and velocity field in 3D

Find $\phi \in \mathrm{H}^2(\Omega)$ and $\boldsymbol{V} \in \mathrm{H}^1(\Omega)^3$ such that

$$-\left(\frac{\partial \boldsymbol{V}_1}{\partial x} + \frac{\partial \boldsymbol{V}_2}{\partial y} + \frac{\partial \boldsymbol{V}_3}{\partial z}\right) = 0 \quad \text{in } \Omega, \tag{4.39}$$

$$\boldsymbol{V}_1 - \frac{\partial \phi}{\partial x} = 0 \quad \text{in } \Omega, \tag{4.40}$$

$$\boldsymbol{V}_2 - \frac{\partial \phi}{\partial y} = 0 \quad \text{in } \Omega, \tag{4.41}$$

$$\boldsymbol{V}_3 - \frac{\partial \phi}{\partial z} = 0 \quad \text{in } \Omega, \tag{4.42}$$

with boundary conditions (4.35) to (4.37).

We can also replace (4.39) by $-\Delta\phi = 0$.

Let $\boldsymbol{w} = \begin{pmatrix} \phi \\ \boldsymbol{V}_1 \\ \boldsymbol{V}_2 \\ \boldsymbol{V}_3 \end{pmatrix}$, the previous PDE can be written as a vector boundary value problem (see section 1.2) where the $\mathcal{H}$-operator is given by

$$\mathcal{H}(\boldsymbol{w}) = 0 \tag{4.43}$$

with

$$\mathcal{H}_{1,1} = 0, \quad \mathcal{H}_{1,2} = \mathcal{L}_{\mathbb{O},-\boldsymbol{e}_1,\boldsymbol{0},0}, \quad \mathcal{H}_{1,3} = \mathcal{L}_{\mathbb{O},-\boldsymbol{e}_2,\boldsymbol{0},0}, \quad \mathcal{H}_{1,4} = \mathcal{L}_{\mathbb{O},-\boldsymbol{e}_3,\boldsymbol{0},0}, \tag{4.44}$$
$$\mathcal{H}_{2,1} = \mathcal{L}_{\mathbb{O},\boldsymbol{0},-\boldsymbol{e}_1,0}, \quad \mathcal{H}_{2,2} = \mathcal{L}_{\mathbb{O},\boldsymbol{0},\boldsymbol{0},1}, \quad \mathcal{H}_{2,3} = 0, \quad \mathcal{H}_{2,4} = 0, \tag{4.45}$$
$$\mathcal{H}_{3,1} = \mathcal{L}_{\mathbb{O},\boldsymbol{0},-\boldsymbol{e}_2,0}, \quad \mathcal{H}_{3,2} = 0, \quad \mathcal{H}_{3,3} = \mathcal{L}_{\mathbb{O},\boldsymbol{0},\boldsymbol{0},1}, \quad \mathcal{H}_{3,4} = 0, \tag{4.46}$$
$$\mathcal{H}_{4,1} = \mathcal{L}_{\mathbb{O},\boldsymbol{0},-\boldsymbol{e}_3,0}, \quad \mathcal{H}_{4,2} = 0, \quad \mathcal{H}_{4,3} = 0, \quad \mathcal{H}_{4,4} = \mathcal{L}_{\mathbb{O},\boldsymbol{0},\boldsymbol{0},1}, \tag{4.47}$$

and $\boldsymbol{e}_1 = (1,0,0)^t$, $\boldsymbol{e}_2 = (0,1,0)^t$, $\boldsymbol{e}_3 = (0,0,1)^t$.

The conormal derivatives are given by

$$\frac{\partial \boldsymbol{w}_1}{\partial n_{\mathcal{H}_{1,1}}} = 0, \quad \frac{\partial \boldsymbol{w}_1}{\partial n_{\mathcal{H}_{2,1}}} = 0, \quad \frac{\partial \boldsymbol{w}_1}{\partial n_{\mathcal{H}_{3,1}}} = 0, \quad \frac{\partial \boldsymbol{w}_1}{\partial n_{\mathcal{H}_{4,1}}} = 0,$$
$$\frac{\partial \boldsymbol{w}_2}{\partial n_{\mathcal{H}_{1,2}}} = \boldsymbol{V}_1\boldsymbol{n}_1, \quad \frac{\partial \boldsymbol{w}_2}{\partial n_{\mathcal{H}_{2,2}}} = 0, \quad \frac{\partial \boldsymbol{w}_2}{\partial n_{\mathcal{H}_{3,2}}} = 0, \quad \frac{\partial \boldsymbol{w}_2}{\partial n_{\mathcal{H}_{4,2}}} = 0,$$
$$\frac{\partial \boldsymbol{w}_3}{\partial n_{\mathcal{H}_{1,3}}} = \boldsymbol{V}_2\boldsymbol{n}_2, \quad \frac{\partial \boldsymbol{w}_3}{\partial n_{\mathcal{H}_{2,3}}} = 0, \quad \frac{\partial \boldsymbol{w}_3}{\partial n_{\mathcal{H}_{3,3}}} = 0, \quad \frac{\partial \boldsymbol{w}_3}{\partial n_{\mathcal{H}_{4,3}}} = 0,$$
$$\frac{\partial \boldsymbol{w}_4}{\partial n_{\mathcal{H}_{1,4}}} = \boldsymbol{V}_3\boldsymbol{n}_3, \quad \frac{\partial \boldsymbol{w}_4}{\partial n_{\mathcal{H}_{2,4}}} = 0, \quad \frac{\partial \boldsymbol{w}_4}{\partial n_{\mathcal{H}_{3,4}}} = 0, \quad \frac{\partial \boldsymbol{w}_4}{\partial n_{\mathcal{H}_{4,4}}} = 0,$$

So we obtain

$$\sum_{\alpha=1}^{4} \frac{\partial \boldsymbol{w}_\alpha}{\partial n_{\mathcal{H}_{1,\alpha}}} = \langle \boldsymbol{V}, \boldsymbol{n} \rangle = \langle \boldsymbol{\nabla}\, \phi, \boldsymbol{n} \rangle, \tag{4.48}$$

and

$$\sum_{\alpha=1}^{4} \frac{\partial \boldsymbol{w}_\alpha}{\partial n_{\mathcal{H}_{2,\alpha}}} = \sum_{\alpha=1}^{4} \frac{\partial \boldsymbol{w}_\alpha}{\partial n_{\mathcal{H}_{3,\alpha}}} = \sum_{\alpha=1}^{4} \frac{\partial \boldsymbol{w}_\alpha}{\partial n_{\mathcal{H}_{4,\alpha}}} = 0. \tag{4.49}$$

From (4.49), we cannot impose boundary conditions on components 2 to 4. Thus, with notation of section 1.2, we have $\Gamma_2^N = \Gamma_3^N = \Gamma_4^N = \Gamma$ with $g_2^N = g_3^N = g_4^N = 0$.

To take into account boundary conditions (4.35) to (4.37), we set $\Gamma_1^D = \Gamma_{1020} \cup \Gamma_{1021} \cup \Gamma_{2020} \cup \Gamma_{2021}$, $\Gamma_1^N = \Gamma \backslash \Gamma_1^D$ and $g_1^D = \delta_{\Gamma_{1020}\cup\Gamma_{2020}} - \delta_{\Gamma_{1021}\cup\Gamma_{2021}}$, $g_1^N = 0$.

The operator in (4.30) is given by $\mathcal{L}_{\alpha\mathbb{I},\boldsymbol{0},\boldsymbol{V},\beta}$. The conormal derivative $\frac{\partial u}{\partial n_{\mathcal{L}}}$ is

$$\frac{\partial u}{\partial n_{\mathcal{L}}} := \langle \mathbb{A}\,\boldsymbol{\nabla}\, u, \boldsymbol{n} \rangle - \langle \boldsymbol{b}u, \boldsymbol{n} \rangle = \alpha \frac{\partial u}{\partial n}.$$

The code using the package for solving (4.39)-(4.42) is given in Listing 4.10

Listing 4.10: **Stationary heat with potential flow in 3D, Octave code (method 2)**

```
d=3;dim=3;m=4;
Hop=Hoperator(dim,d,m);
Hop.set(1,2,Loperator(dim,d,[],{−1,0,0},[],[]));
Hop.set(1,3,Loperator(dim,d,[],{0,−1,0},[],[]));
Hop.set(1,4,Loperator(dim,d,[],{0,0,−1},[],[]));
Hop.set(2,1,Loperator(dim,d,[],[],{−1,0,0},[]));
Hop.set(2,2,Loperator(dim,d,[],[],[],1));
Hop.set(3,1,Loperator(dim,d,[],[],{0,−1,0},[]));
Hop.set(3,3,Loperator(dim,d,[],[],[],1));
Hop.set(4,1,Loperator(dim,d,[],[],{0,0,−1},[]));
Hop.set(4,4,Loperator(dim,d,[],[],[],1));
bvpFlow=BVP(Th,PDEelt(Hop));
bvpFlow.setDirichlet(1020,−1,1);
bvpFlow.setDirichlet(1021,1,1);
```

```
bvpFlow.setDirichlet(2020,−1,1);
bvpFlow.setDirichlet(2021,1,1);
W=bvpFlow.solve('split',true);
af=@(x,y,z) 1+(z−1).^2;
Lop=Loperator(dim,d,{af,[],[];[],af,[];[],[],af},[],{W{2},W{3},W{4}},0.01);
bvpHeat=BVP(Th,PDEelt(Lop));
bvpHeat.setDirichlet(1020,30.);
bvpHeat.setDirichlet(2020,30.);
bvpHeat.setDirichlet(10, @(x,y,z) 10∗(abs(z−1)>0.5));
U=bvpHeat.solve();
```

Bibliography

[1] F. Cuvelier and G. Scarella. A generic way to solve partial differential equations by the $\mathbb{P}_1$-Lagrange finite element method in vector languages. `https://www.math.univ-paris13.fr/~cuvelier/software/docs/Recherch/VecFEM/distrib/0.1b1/vecFEMP1_report-0.1b1.pdf`, 2015.

[2] François Cuvelier, Caroline Japhet, and Gilles Scarella. An efficient way to assemble finite element matrices in vector languages. *BIT Numerical Mathematics*, 56(3):833–864, dec 2015.

[3] G. Dhatt, E. Lefrançois, and G. Touzot. *Finite Element Method*. Wiley, 2012.

[4] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*, volume 23 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1994.