fc-vfem$\mathbb{P}_1$-eigs Matlab toolbox, User's Guide[1]

Eigenvalues addon to the ...

version 0.2.2

François Cuvelier[2]

Sunday 5th April, 2020

[1] LaTeX manual, revision 0.1.2, compiled with Matlab 2019a, and toolboxes `fc-vfemp1`[0.2.2], `fc-tools`[0.0.32], `fc-bench`[0.1.2], `fc-hypermesh`[1.0.3], `fc-amat`[0.1.2], `fc-meshtools`[0.1.3], `fc-graphics4mesh`[0.1.4], `fc-oogmsh`[0.2.3], `fc-siplt`[0.2.3], `fc-simesh`[0.4.3]

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

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

Contents

Chapter 1

Generic Boundary Value Problems

The notations of [6] 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 $\mathbb{R}^d$, $d \geq 1$. 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 [6] 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

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

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

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

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

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

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 vfem$\mathbb{P}_1$-eigs toolbox, 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

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

where Ω and its boundaries are given in Figure 1.1.

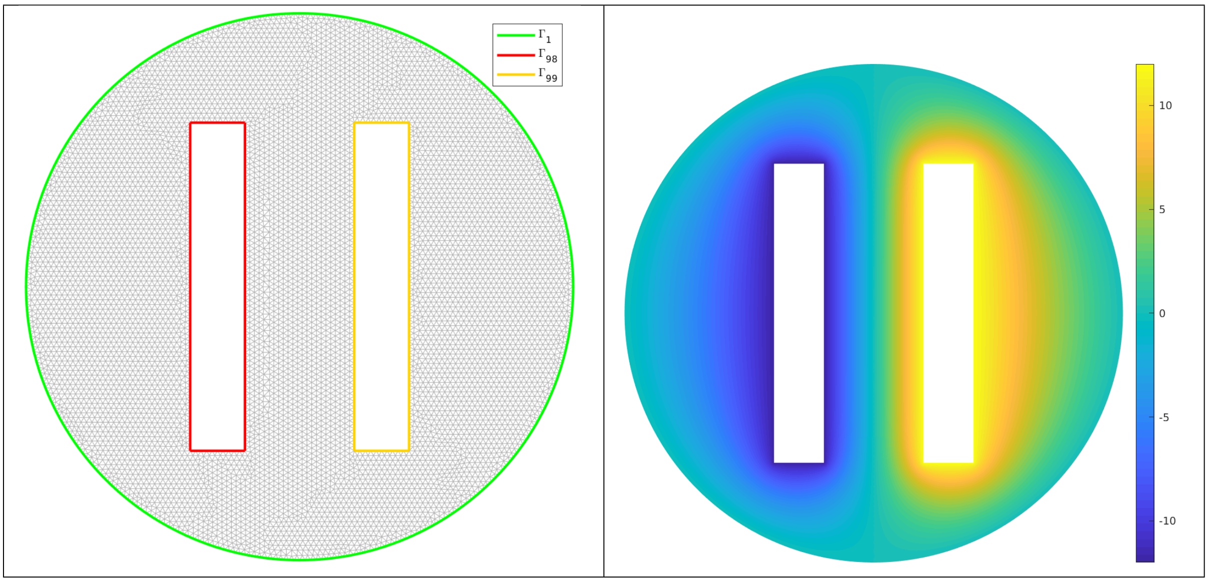


Figure 1.1: 2D condenser mesh and boundaries (left) and numerical solution (right)

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 1.1 a complete code is given to solve this problem and in Table 1.1 computational times for assembling and solving steps are given with various size meshes.

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

```
meshfile=fc_oogmsh.buildmesh2d('condenser',10); % generate mesh
Th=fc_simesh.siMesh(meshfile); % read mesh
Lop=fc_vfemp1.Loperator(2,2,{1,0;0,1},[],[],[]);
pde=fc_vfemp1.PDE(Lop);
bvp=fc_vfemp1.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;axis off,axis image;
Th.plotmesh('d',1,'Linewidth',2,'inlegend',true)
legend('show')
figure(2)
Th.plot(U,'edgecolor','none','facecolor','interp')
axis off,axis image;colorbar
```

N	$\mathrm{n_q}$	$\mathrm{n_{me}}$	Assembly	Solve
100	10 201	20 000	0.129 (s)	0.029 (s)
200	40 401	80 000	0.187 (s)	0.129 (s)
300	90 601	180 000	0.404 (s)	0.415 (s)
400	160 801	320 000	0.705 (s)	0.687 (s)
500	251 001	500 000	1.103 (s)	1.128 (s)
600	361 201	720 000	1.673 (s)	1.836 (s)
700	491 401	980 000	2.283 (s)	2.799 (s)
800	641 601	1 280 000	3.153 (s)	3.962 (s)

Table 1.1: Computational times for assembling and solving the 2D condenser BVP, described in *Scalar* BVP 2, with various size meshes.

1.2 Vector boundary value problem

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

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

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

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

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 vfem$\mathbb{P}_1$-eigs toolbox, 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

$$-\Delta u_1 + u_2 = 0 \quad \text{in } \Omega \subset \mathbb{R}^2, \tag{1.18}$$

$$-\Delta u_2 + u_1 = 0 \quad \text{in } \Omega \subset \mathbb{R}^2, \tag{1.19}$$

$$(u_1, u_2) = (0, 0) \quad \text{on } \Gamma_1, \tag{1.20}$$

$$(u_1, u_2) = (-12., +12.) \quad \text{on } \Gamma_{98}, \tag{1.21}$$

$$(u_1, u_2) = (+12., -12.) \quad \text{on } \Gamma_{99}, \tag{1.22}$$

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 1.2 a complete code is given to solve this problem. Numerical solutions are represented in Figure 1.2. In Table 1.2 computational times for assembling and solving steps are given with various size meshes.

Listing 1.2: Complete Matlab code to solve the simple 2D vector BVP with graphical representations

```
meshfile=fc_oogmsh.buildmesh2d('condenser',10); % generate mesh
Th=fc_simesh.siMesh(meshfile); % read mesh
Hop=fc_vfemp1.Hoperator(2,2,2);
Hop.set([1,2],[1,2],fc_vfemp1.Loperator(2,2,{1,[];[],1},[],[],[]));
Hop.set([1,2],[2,1],fc_vfemp1.Loperator(2,2,[],[],[],1));
pde=fc_vfemp1.PDE(Hop);
bvp=fc_vfemp1.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
```

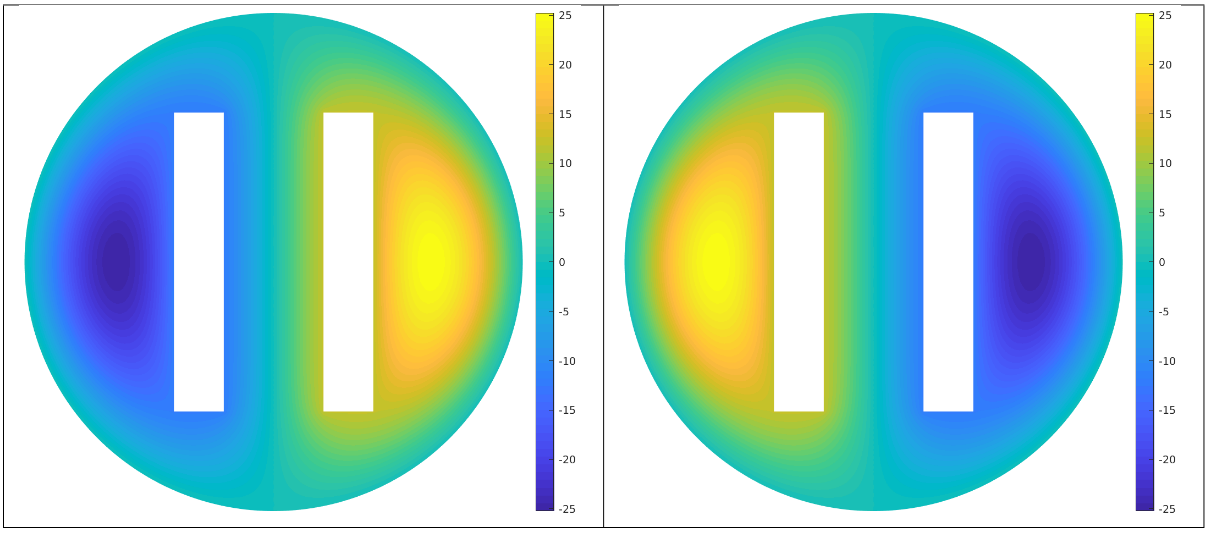


Figure 1.2: 2D simple vector BVP, u_1 numerical solution (left) and u_2 numerical solution (right)

N	n_q	n_{me}	n_{dof}	Assembly	Solve
10	8 151	15 708	16 302	0.197 (s)	0.122 (s)
20	31 742	62 296	63 484	0.426 (s)	0.569 (s)
30	70 744	139 706	141 488	1.110 (s)	1.440 (s)
40	124 930	247 484	249 860	2.138 (s)	2.524 (s)
50	194 775	386 580	389 550	3.703 (s)	4.744 (s)
60	279 962	556 360	559 924	5.671 (s)	7.402 (s)

Table 1.2: Computational times for assembling and solving the 2D simple vector BVP, described in *Scalar* BVP 2, with various size meshes.

Chapter 2

Generalized Eigenvalue scalar BVP

We want to solve generalized eigenvalue problems coming from scalar BVP's.

The **generalized eigenvalue problem** associated with *scalar* BVP (1.2)-(1.4) can be written as

***Scalar* EBVP 1 : generic problem**

Find $(\lambda, u) \in \mathbb{K} \times \mathrm{H}^2(\Omega)$ such that

$$\begin{aligned} \mathcal{L}(u) =& \lambda\mathcal{B}(u) && \text{in } \Omega, && (2.1) \\ u =& 0 && \text{on } \Gamma^D, && (2.2) \\ \frac{\partial u}{\partial n_{\mathcal{L}}} + a^R u =& 0 && \text{on } \Gamma^R. && (2.3) \end{aligned}$$

where $\mathcal{B} = \mathcal{L}_{\mathbb{O}_{d\times d}, \mathbf{0}_d, \tilde{\boldsymbol{c}}, \tilde{a_0}}$.

We briefly describe the main function that will be used to solve eigenvalues problems. Let bvp be a fc_vfemp1.BVP object.

2.1 fc_vfemp1.addon.eigs.solve function

The function fc_vfemp1.addon.eigs.solve returns a few eigenvalues and eigenvectors obtained by solving a generalized eigenvalue scalar BVP with $\mathbb{P}_1$-Lagrange finite elements.

```
[U,lambda]=fc_vfemp1.addon.eigs.solve(bvp)
[U,lambda]=fc_vfemp1.addon.eigs.solve(bvp,Name,Value)
```

Description

[U,lambda]=fc_vfemp1.addon.eigs.solve(bvp) calculates 6 of eigenvalues and eigenvectors/eigenfunctions of the generalized eigenvalue scalar BVP given by bvp a fc_vfemp1.BVP object which described the *scalar* BVP (1.2)-(1.4) with all right-hand sides equal to zeros, i.e. $f := 0$, $g^D := 0$ and $g^R := 0$.

By default the $\mathcal{B}$ is the operator $\mathcal{L}_{\mathbb{O}_{d\times d},\mathbf{0}_d,\mathbf{0}_d,1}$.
The outputs are those given by the eigs Matlab function:

- lambda is a k-by-1 vector where lambda(i) is the i-th eigenvalue approximation and k is the number of eigenvalues computed (6 by default).
- U is a N-by-k matrix where k is the number of eigenvalues and U(:,i) is the eigenvector ($\mathbb{P}_1$-Lagrange finite elements approximation) associated with the eigenvalue lambda(i).

[U,lambda]=fc_vfemp1.addon.eigs.solve(bvp,Name,Value) specifies function options using one or more Name,Value pair.

- 'Bop' : to specify a fc_vfemp1.Loperator object corresponding to the operator $\mathcal{B}$. The default operator is $\mathcal{L}_{\mathbb{O}_{d\times d},\mathbf{0}_d,\mathbf{0}_d,1}$ for scalar BVP.
- 'neigs' : to specify the number of eigenvalues and eigenvectors to be computed. Default is 6.
- 'sigma' : to specify the sigma parameter of the eigs Matlab function. Default is 0.

2.2 2D samples

2.2.1 2D Laplace eigenvalues problem with Dirichlet boundary condition

We want to solve the eigenvalue problem given by (2.4)-(2.5)).

Usual EBVP 1 : 2D Laplace with Dirichlet boundary condition

Find $(\lambda, u) \in \mathbb{C} \times \mathrm{H}^2(\Omega)$ such that

$$-\Delta u = \lambda u \quad \text{in } \Omega, \tag{2.4}$$

$$u = 0 \quad \text{on } \Gamma, \tag{2.5}$$

The problem (2.4)-(2.5) can be equivalently write as the *Scalar* EBVP 1:

Scalar EBVP 2 : 2D Laplace with Dirichlet boundary condition

Find $(\lambda, u) \in \mathbb{K} \times \mathrm{H}^2(\Omega)$ such that

$$\begin{aligned} \mathcal{L}(u) &= \lambda\mathcal{B}(u) && \text{in } \Omega, \\ u &= 0 && \text{on } \Gamma^D = \Gamma, \end{aligned}$$

where $\mathcal{L} = \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}$, $\mathcal{B} = \mathcal{L}_{\mathbb{O},\mathbf{0},\mathbf{0},1}$.

Application on the rectangle $\Omega = [0, L] \times [0, H]$.

The eigenvalues and the associated eigen functions are given by

$$\lambda_{k,l} = \left(\frac{k\pi}{L}\right)^2 + \left(\frac{l\pi}{H}\right)^2, \quad u_{k,l}(x, y) = \sin(\frac{k\pi}{L}x)\sin(\frac{l\pi}{H}y), \quad \forall (k, l) \in \mathbb{N}^* \times \mathbb{N}^*. \tag{2.6}$$

In Table 2.1 and Table 2.2, the first eigenvalues are given recpectively for $L = 1$, $H = 1$ and for $L = 2$, $H = 3$.

$l \backslash k$	1	2	3	4	5
1	$2\,\pi^2 \approx 19.7392$	$5\,\pi^2 \approx 49.3480$	$10\,\pi^2 \approx 98.6961$	$17\,\pi^2 \approx 167.783$	$26\,\pi^2 \approx 256.610$
2	$5\,\pi^2 \approx 49.3480$	$8\,\pi^2 \approx 78.9568$	$13\,\pi^2 \approx 128.305$	$20\,\pi^2 \approx 197.392$	$29\,\pi^2 \approx 286.219$
3	$10\,\pi^2 \approx 98.6961$	$13\,\pi^2 \approx 128.305$	$18\,\pi^2 \approx 177.653$	$25\,\pi^2 \approx 246.740$	$34\,\pi^2 \approx 335.567$
4	$17\,\pi^2 \approx 167.783$	$20\,\pi^2 \approx 197.392$	$25\,\pi^2 \approx 246.740$	$32\,\pi^2 \approx 315.827$	$41\,\pi^2 \approx 404.654$
5	$26\,\pi^2 \approx 256.610$	$29\,\pi^2 \approx 286.219$	$34\,\pi^2 \approx 335.567$	$41\,\pi^2 \approx 404.654$	$50\,\pi^2 \approx 493.480$

Table 2.1: Eingenvalues $\lambda_{k,l}$ for $(k,l) \in [\![1,5]\!]$ with $L = 1$, $H = 1$

$l \backslash k$	1	2	3	4	5
1	$\frac{13}{36}\,\pi^2 \approx 3.56402$	$\frac{25}{36}\,\pi^2 \approx 6.85389$	$\frac{5}{4}\,\pi^2 \approx 12.3370$	$\frac{73}{36}\,\pi^2 \approx 20.0134$	$\frac{109}{36}\,\pi^2 \approx 29.8830$
2	$\frac{10}{9}\,\pi^2 \approx 10.9662$	$\frac{13}{9}\,\pi^2 \approx 14.2561$	$2\,\pi^2 \approx 19.7392$	$\frac{25}{9}\,\pi^2 \approx 27.4156$	$\frac{34}{9}\,\pi^2 \approx 37.2852$
3	$\frac{85}{36}\,\pi^2 \approx 23.3032$	$\frac{97}{36}\,\pi^2 \approx 26.5931$	$\frac{13}{4}\,\pi^2 \approx 32.0762$	$\frac{145}{36}\,\pi^2 \approx 39.7526$	$\frac{181}{36}\,\pi^2 \approx 49.6222$
4	$\frac{37}{9}\,\pi^2 \approx 40.5750$	$\frac{40}{9}\,\pi^2 \approx 43.8649$	$5\,\pi^2 \approx 49.3480$	$\frac{52}{9}\,\pi^2 \approx 57.0244$	$\frac{61}{9}\,\pi^2 \approx 66.8940$
5	$\frac{229}{36}\,\pi^2 \approx 62.7817$	$\frac{241}{36}\,\pi^2 \approx 66.0715$	$\frac{29}{4}\,\pi^2 \approx 71.5546$	$\frac{289}{36}\,\pi^2 \approx 79.2310$	$\frac{325}{36}\,\pi^2 \approx 89.1006$

Table 2.2: Eingenvalues $\lambda_{k,l}$ for $(k,l) \in [\![1,5]\!]$ with $L = 2$, $H = 3$

The Listing 2.1 is part of the Matlab script: fc_vfemp1.addon.eigs.demos.dim2d.Laplacian01. It contains the complete code computing the first smallest magnitude eight eigenvalues with their graphic representations. These graphic results ar given in Figure 2.1.

```
% Setting mesh:
Th=fc_simesh.hypercube(2,[L*N,H*N],'trans',@(q) [L*q(1,:);H*q(2,:)]);

% Setting eBVP:
Lop=fc_vfemp1.Loperator(2,2,{1,0;0,1},[],[],[]);
pde=fc_vfemp1.PDE(Lop);
bvp=fc_vfemp1.BVP(Th,pde);
for lab=1:4, bvp.setDirichlet( lab, 0);end

% Solving eBVP:
[eVec,lambda]=fc_vfemp1.addon.eigs.solve(bvp,'neigs',neigs,'sigma',sigma);

% Plotting of eigenfunctions:
[eVec,lambda]=fc_vfemp1.addon.eigs.splitsort(bvp,eVec,lambda);
fc_vfemp1.addon.eigs.plot(Th,eVec,lambda);
```

Listing 2.1: 2D Laplace eigenvalues problem with Dirichlet boundary condition $\Omega = [0, L] \times [0, H]$, obtained by using fc_vfemp1.addon.eigs.demos.dim2d.Laplacian01 script with L=2, H=3, N=150, neigs=8 and sigma=0.

We represent in Figure 2.2 the order of convergence of the first ten eigenvalues.

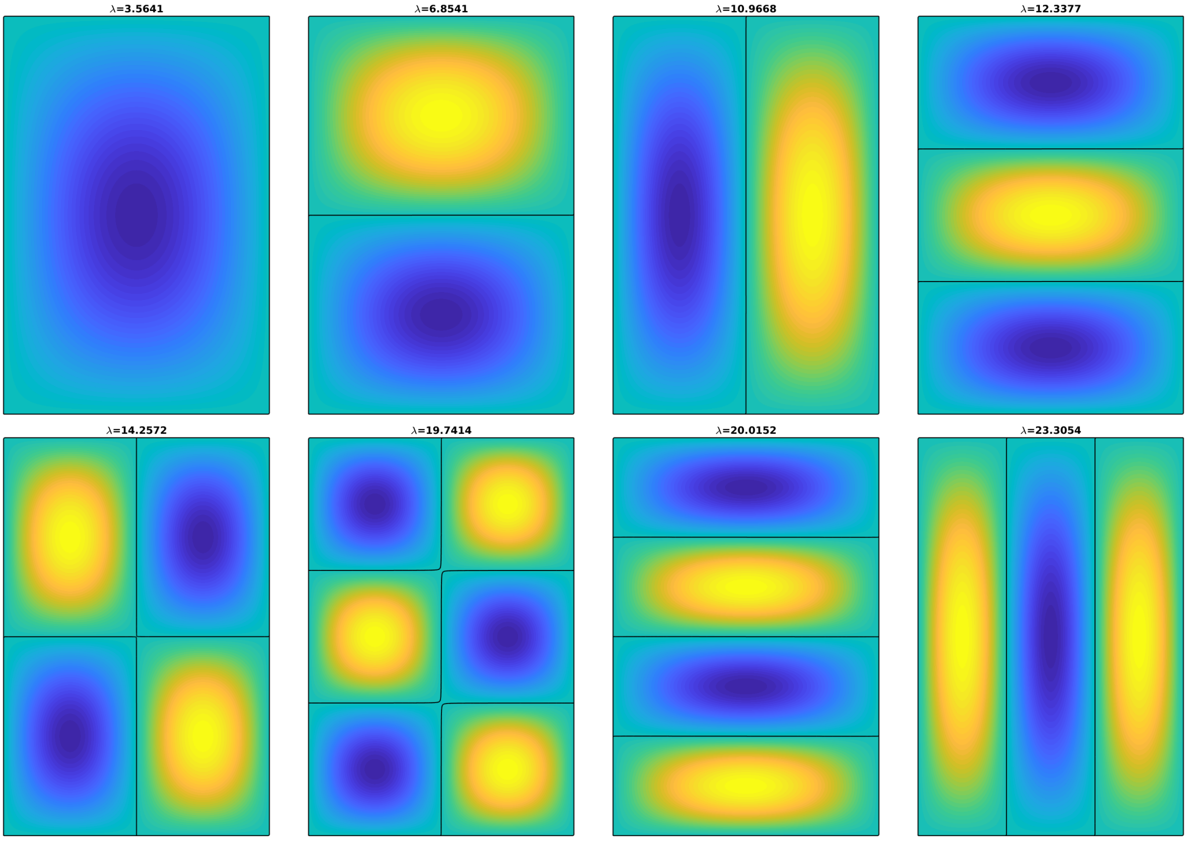


Figure 2.1: 2D Laplace in rectangle $[0, 2] \times [0, 3]$ with Dirichlet boundary conditions : eigenfunctions of the smallest magnitude eigenvalues

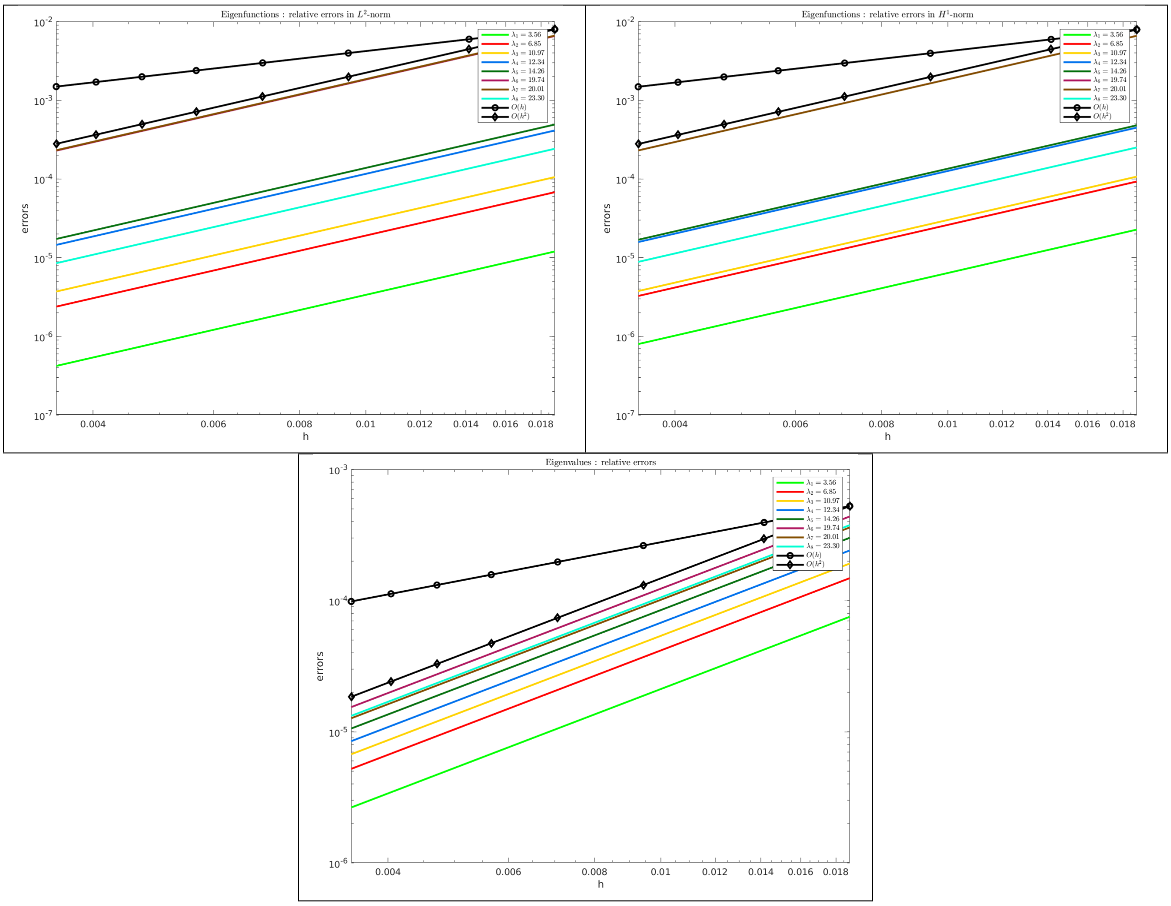


Figure 2.2: eigenvalues and eigenfunctions : order computation for 2D Laplace with Dirichlet boundary condition on rectangle (regular meshes) . Relative errors of eigenfunctions in L^2-norm (upper left) and H^1-norm (upper right). Relative errors of eigenvalues (bottom).

One can see that a superconvergence phenomena occurs due to regularity of the hypercube mesh. Indeed, for the H^1-norm an order 1 is expected. To hightlight it, gmsh is now used to generate all the meshes of Ω : results are given in Figure 2.3.

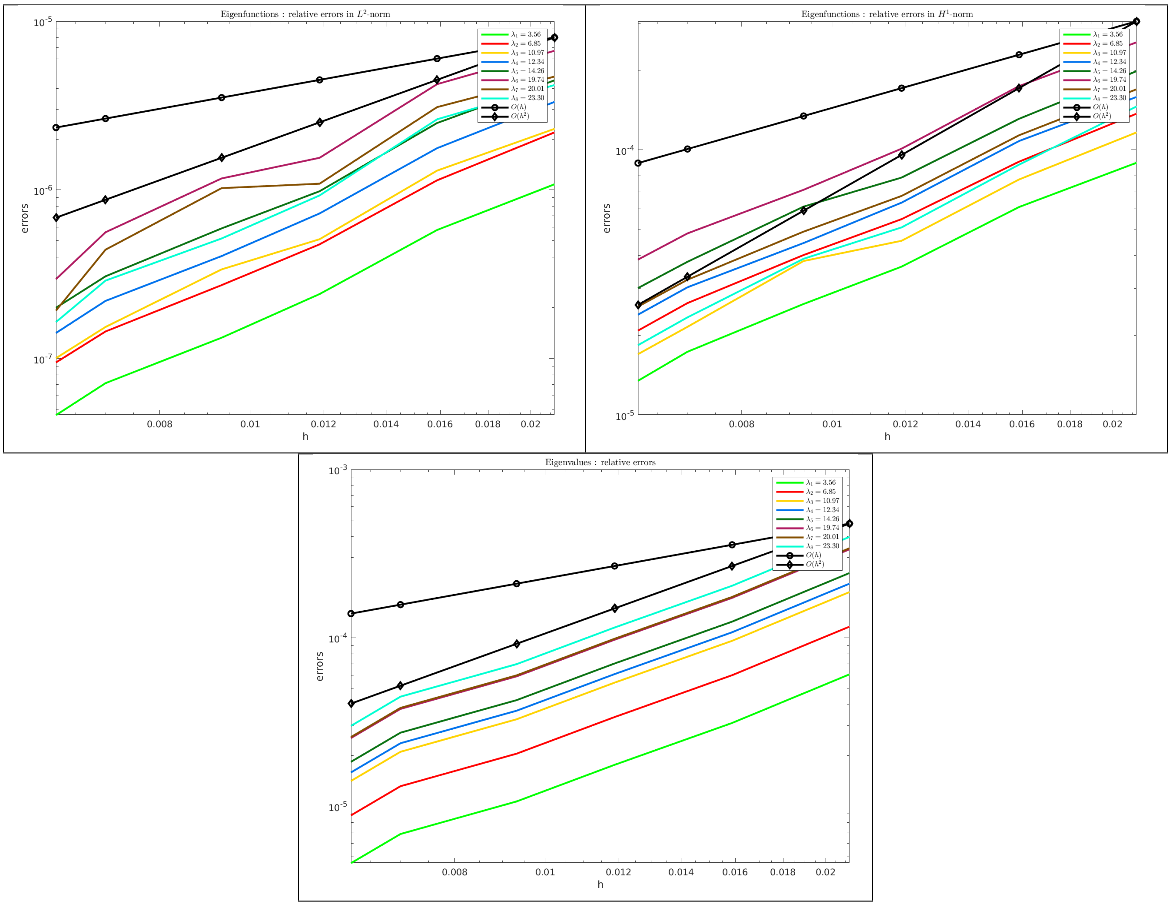


Figure 2.3: eigenvalues and eigenfunctions : order computation for 2D Laplace with Dirichlet boundary condition on rectangle (gmsh meshes). Relative errors of eigenfunctions in L^2-norm (upper left) and H^1-norm (upper right). Relative errors of eigenvalues (bottom).

Application on the unit disk.

Let $\Omega \subset \mathbb{R}^2$ be the unit disk meshed by gmsh and given in Figure 2.4.

Let α_{nl} bet the l-th zero of the Bessel function of the first kind J_n. The eigenvalues are given by

$$\lambda_{n,l} = \alpha_{nl}^2 \quad \forall (n,l) \in \mathbb{N} \times \mathbb{N}^*$$

In Table 2.3, the values of α_{nl} are given for $(n,l) \in [\![0,4]\!][\![1,5]\!]$.

The eigenvalues are simple for $n = 0$ and twice degenerate for $n > 0$.

l	J_0	J_1	J_2	J_3	J_4	J_5	J_6
1	2.4048256	3.8317060	5.1356223	6.3801619	7.5883424	8.7714838	9.9361095
2	5.5200781	7.0155867	8.4172441	9.7610231	11.064709	12.338604	13.589290
3	8.6537279	10.173468	11.619841	13.015201	14.372537	15.700174	17.003820
4	11.791534	13.323692	14.795952	16.223466	17.615966	18.980134	20.320789
5	14.930918	16.470630	17.959819	19.409415	20.826933	22.217800	23.586084
6	18.071064	19.615859	21.116997	22.582730	24.019020	25.430341	26.820152

Table 2.3: Zeros of the Bessel function of the first kind J_n

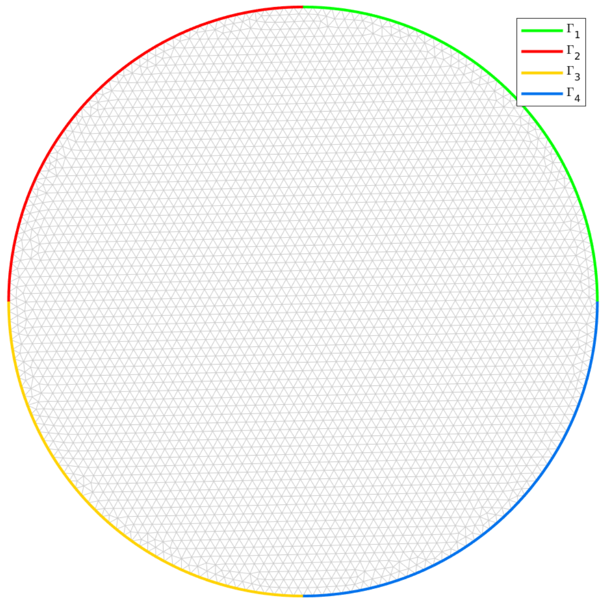


Figure 2.4: Unit disk with four boundaries

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
5.7831860	14.681971	14.681971	26.374616	26.374616	30.471262	40.706466	40.706466
λ_9	λ_{10}	λ_{11}	λ_{12}	λ_{13}	λ_{14}	λ_{15}	λ_{16}
49.218456	49.218456	57.582941	57.582941	70.849999	70.849999	74.887007	76.938928
λ_{17}	λ_{18}	λ_{19}	λ_{20}	λ_{21}	λ_{22}	λ_{23}	λ_{24}
76.938928	95.277573	95.277573	98.726272	98.726272	103.49945	103.49945	122.42780

Table 2.4: twenty four first eigenvalues

The Listing 2.2 is part of the Matlab script: fc_vfemp1.addon.eigs.demos.dim2d.Laplacian02. It contains the complete code computing the first smallest magnitude twenty-four eigenvalues with their graphic representations. These graphic results ar given in Figure 2.5.

```
% Setting mesh:
fullgeofile=fc_vfemp1.addon.eigs.get_geo(2,2,'disk4bounds.geo');
meshfile=fc_oogmsh.buildmesh2d(fullgeofile,N);
Th=fc_simesh.siMesh(meshfile);
Th.info('verbose',true)

% Setting eBVP:
Lop=fc_vfemp1.Loperator(2,2,{1,0;0,1},[],[],[]);
pde=fc_vfemp1.PDE(Lop);
bvp=fc_vfemp1.BVP(Th,pde);
for lab=Th.get_labels(1), bvp.setDirichlet( lab, 0);end

% Solving eBVP:
[eVec,lambda]=fc_vfemp1.addon.eigs.solve(bvp,'neigs',neigs,'sigma',sigma);

% Plotting eigenfunctions:
[eVec,lambda]=fc_vfemp1.addon.eigs.splitsort(bvp,eVec,lambda);
fc_vfemp1.addon.eigs.plot(Th,eVec,lambda);
```

Listing 2.2: 2D Laplace eigenvalues problem on unit disk with Dirichlet boundary conditions, obtained by using fc_vfemp1.addon.eigs.demos.dim2d.Laplacian02 script with N=150, neigs=24 and sigma=0

Application on the L-shape domain. The *Lshape* domain Ω meshed by gmsh is given in Figure 2.6.

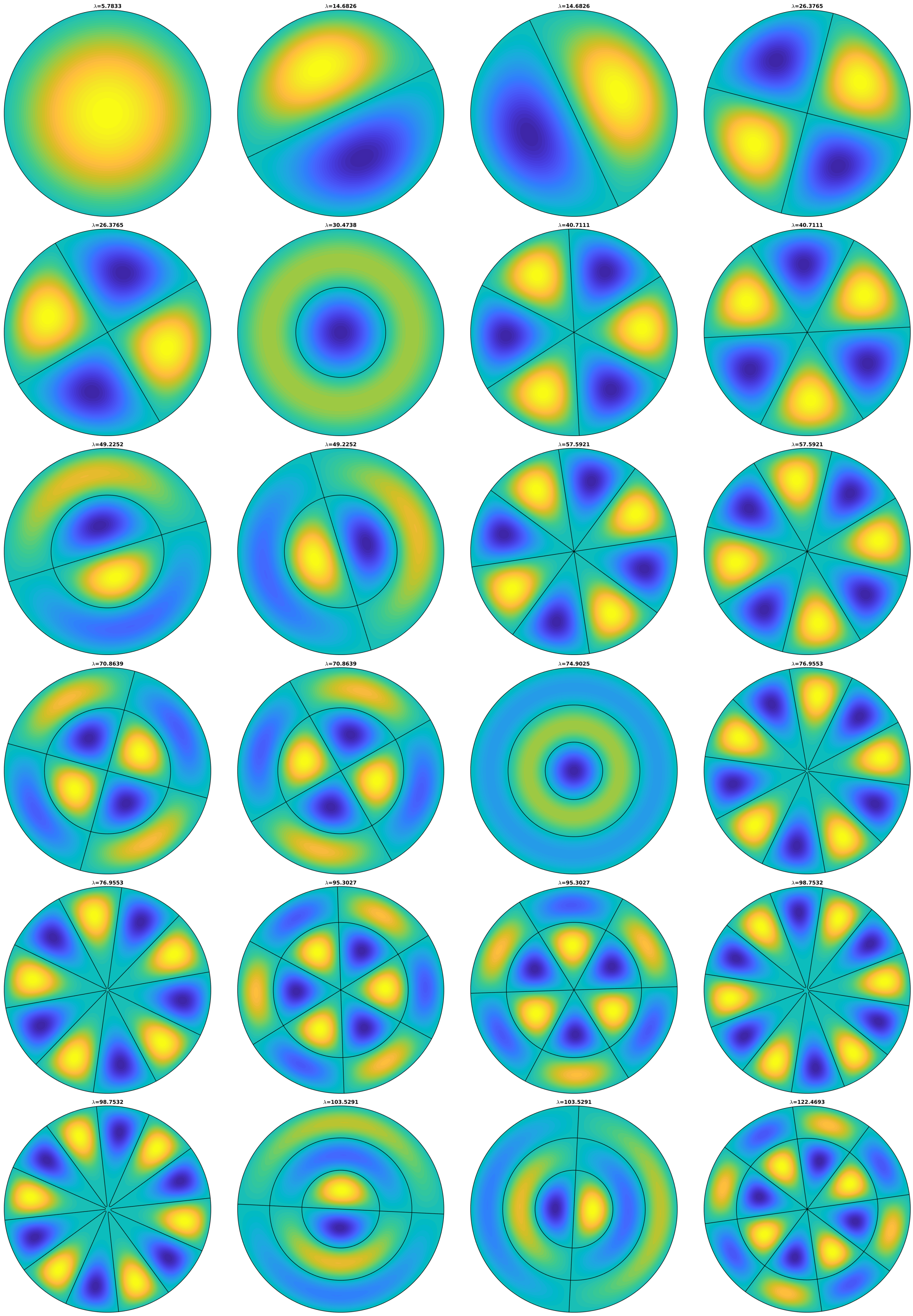


Figure 2.5: 2D Laplace in unit disk with Dirichlet boundary conditions : eigenfunctions of the smallest magnitude eigenvalues

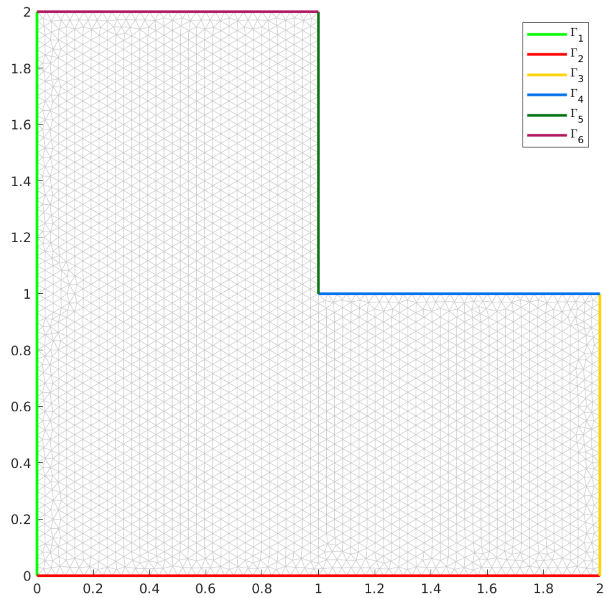


Figure 2.6: *Lshape* domain with four boundaries

To compute the eigenvalues and the eigenfunctions of the Laplacian with Dirichlet boundary condition using $\mathbb{P}_1$-Lagrange finite elements one can used the Matlab script fc_vfemp1.addon.eigs.demos.dim2d.Laplacian03. Part of the source code is given in Listing 2.3

```
% Setting mesh:
fullgeofile=fc_vfemp1.addon.eigs.get_geo(2,2,'Lshape.geo');
meshfile=fc_oogmsh.buildmesh2d(fullgeofile,N);
Th=fc_simesh.siMesh(meshfile);
Th.info('verbose',true)

% Setting eBVP:
Lop=fc_vfemp1.Loperator(2,2,{1,0;0,1},[],[],[]);
pde=fc_vfemp1.PDE(Lop);
bvp=fc_vfemp1.BVP(Th,pde);
for lab=Th.get_labels(1), bvp.setDirichlet( lab, 0);end

% Solving eBVP:
[eVec,lambda]=fc_vfemp1.addon.eigs.solve(bvp,'neigs',neigs,'sigma',sigma);

% Plotting eigenfunctions:
[eVec,lambda]=fc_vfemp1.addon.eigs.splitsort(bvp,eVec,lambda);
fc_vfemp1.addon.eigs.plot(Th,eVec,lambda);
```

Listing 2.3: 2D Laplace eigenvalues problem with Dirichlet boundary condition on *Lshape* domain

Results can be found in [7], Figure 1 page 4 and [3]. From [4] section 6.52 page 122 or [8] Table 1 page 1088, we have the bounds to the first ten eigenvalues of the L-shaped Laplacian problem recalled in Table 2.5. We also give the computed values from a L-shaped mesh with $\mathrm{n_q} = 78980$, $\mathrm{n_{me}} = 156758$ and $h \approx 0.0086192473$.

n	bounds of λ_n from [8]	computed here
1	9.63972384^{44}_{04}	9.6420953
2	15.19725192^{66}_{59}	15.197889
3	$2\pi^2 = 19.739208802178$	19.740287
4	29.52148111^{42}_{38}	29.523893
5	31.9126359^{59}_{37}	31.920617
6	41.4745098^{92}_{66}	41.483163
7	44.9484877^{82}_{77}	44.954078
8	$5\pi^2 = 49.34802200544$	49.354768
9	$5\pi^2 = 49.34802200544$	49.354769
10	56.7096098^{90}_{18}	56.722843

Table 2.5: Bounds to the first ten eigenvalues of the L-shaped Laplacian problem

We represent in Figure 2.7 eigenvectors associated to the first twenty-four smallest magnitude eigenvalues.

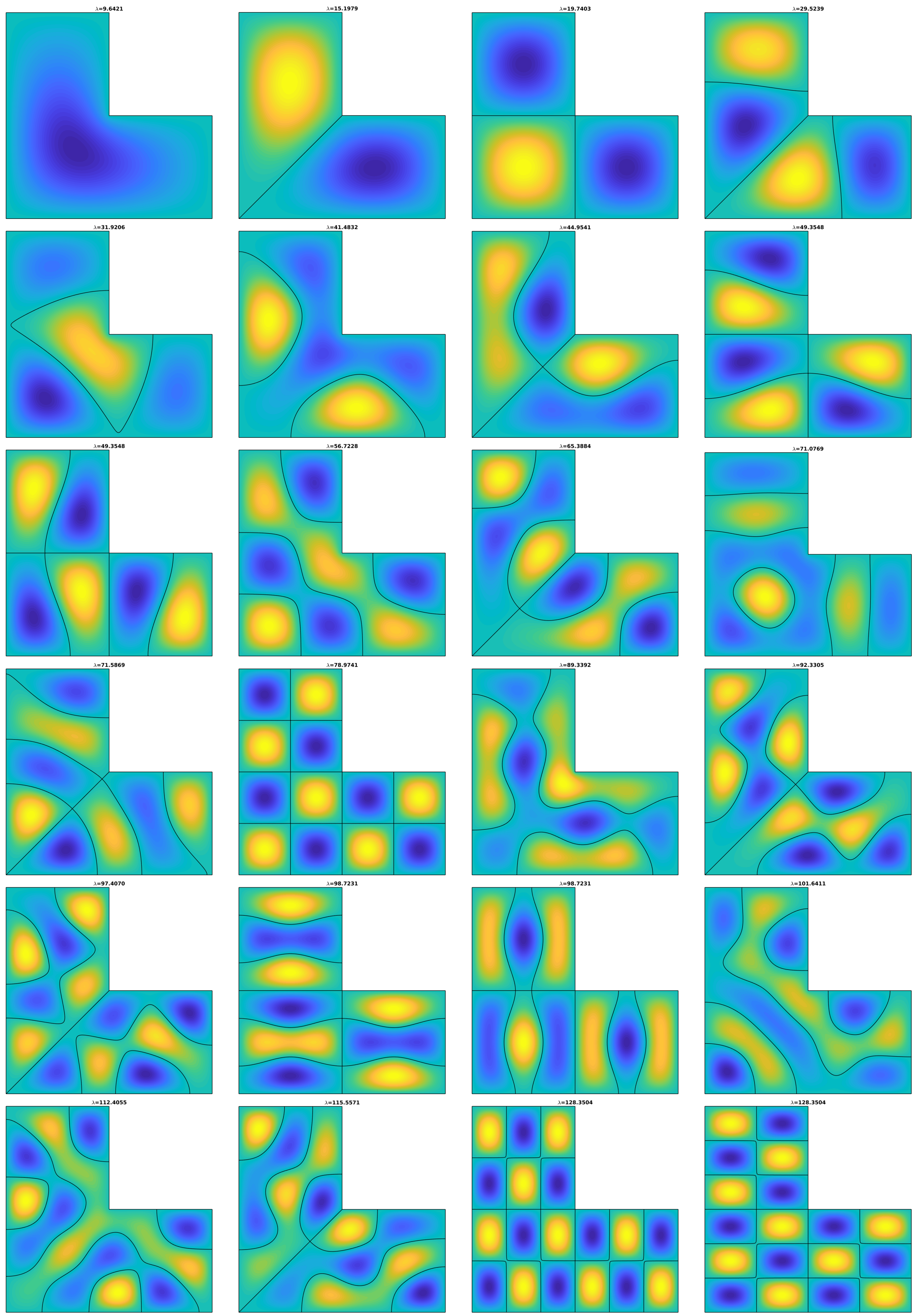


Figure 2.7: 2D Laplace in L-shaped domain with Dirichlet boundary conditions : eigenfunctions of the smallest magnitude eigenvalues

In Figure 2.8 the eigenvectors associated with the four eigenvalues nearest 250 (multiplicity 1) and 493 (multiplicity 3) are represented. This is done by setting *sigma* option to 250 for the first case and to 493 for the second one.

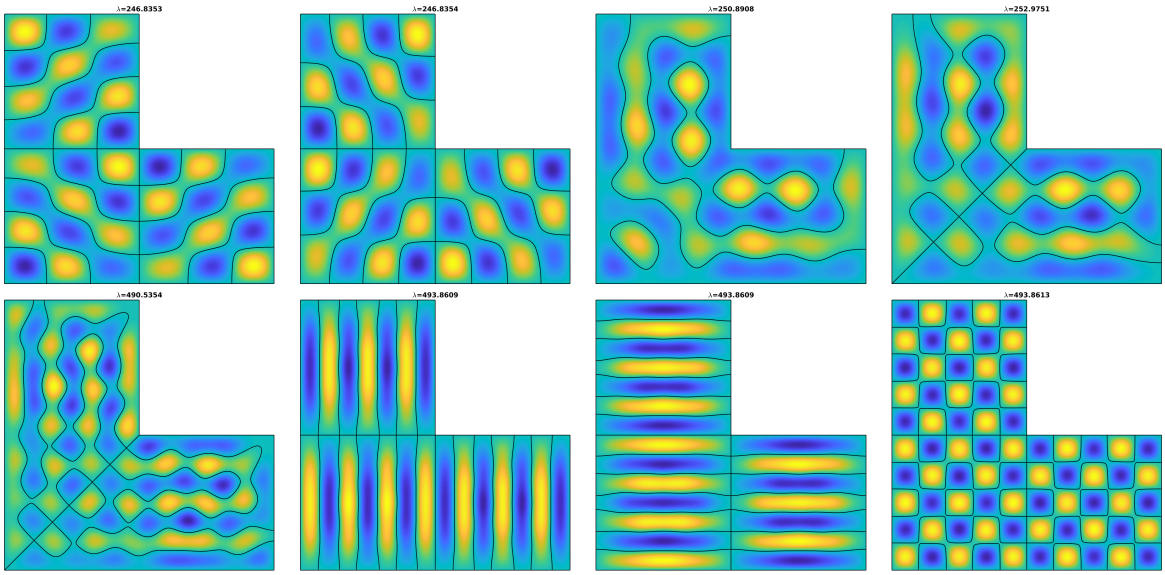


Figure 2.8: 2D Laplace with Dirichlet boundary conditions : eigenfunctions of the eigenvalues near $\lambda_{50} = 250.78548$ (multiplicity 1) and $\lambda_{104} = 493.48022$ (multiplicity 3)

2.2.2 2D Laplace eigenvalues problem with mixed boundary conditions

We want to solve the eigenvalue problem given by (2.7)-(2.10)).

Usual EBVP 2 : 2D Laplace with mixed boundary condition

Find $(\lambda, u) \in \mathbb{K} \times \mathrm{H}^2(\Omega)$ such that

$$-\Delta u = \lambda u \quad \text{in } \Omega, \tag{2.7}$$

$$\frac{\partial u}{\partial n} + \alpha u = 0 \quad \text{on } \Gamma^a, \tag{2.8}$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma^b, \tag{2.9}$$

$$u = 0 \quad \text{on } \Gamma^c, \tag{2.10}$$

The problem (2.7)-(2.10) can be equivalently written as the *Scalar* EBVP 1:

Scalar EBVP 3 : 2D Laplace with mixed boundary condition

Find $(\lambda, u) \in \mathbb{K} \times \mathrm{H}^2(\Omega)$ such that

$$\mathcal{L}(u) = \lambda \mathcal{B}(u) \quad \text{in } \Omega,$$

$$\frac{\partial u}{\partial n_{\mathcal{L}}} + a^R u = 0 \quad \text{on } \Gamma^R = \Gamma^a \cup \Gamma^b,$$

$$u = 0 \quad \text{on } \Gamma^D = \Gamma^c,$$

where $\mathcal{L} = \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}$ (and then $\frac{\partial u}{\partial n_{\mathcal{L}}} = \frac{\partial u}{\partial n}$), $\mathcal{B} = \mathcal{L}_{\mathbb{O},\mathbf{0},\mathbf{0},1}$, $a^R = \alpha\delta_{\Gamma^b}$

Application on the disk with 5 holes domain. Let Γ_1 be the unit disk, Γ_{10} be the disk with center point $(0,0)$ and radius 0.3. Let Γ_{20}, Γ_{21}, Γ_{22} and Γ_{23} be the disks 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 disks (see Figure 2.9).

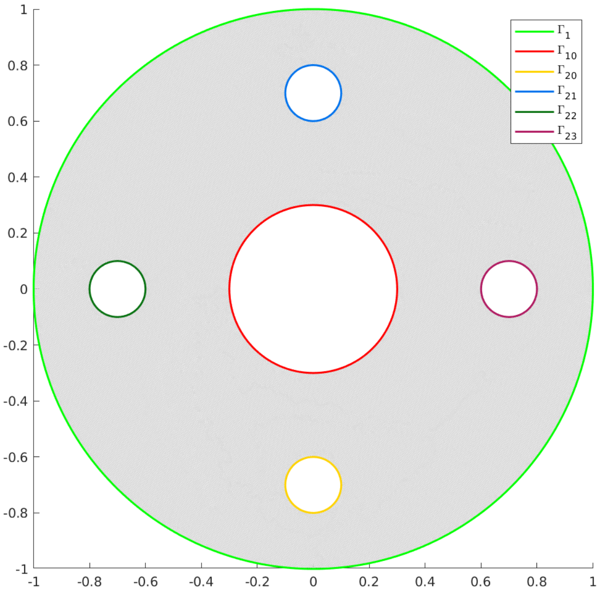


Figure 2.9: Domain and boundaries

We want to solve the eigenvalue problem given by (2.11)-(2.14).

***Scalar* EBVP 4 : 2D Laplace eigenvalues problem with mixed boundary conditions**

Find $(\lambda, u) \in \mathbb{K} \times \mathrm{H}^2(\Omega)$ such that

$$-\Delta u = \lambda u \quad \text{in } \Omega, \tag{2.11}$$

$$\frac{\partial u}{\partial n} + \alpha u = 0 \quad \text{on } \Gamma_{22} \cup \Gamma_{23}. \tag{2.12}$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma_{20} \cup \Gamma_{21}, \tag{2.13}$$

$$u = 0 \quad \text{on } \Gamma_1 \cup \Gamma_{10}. \tag{2.14}$$

So we have, $\Gamma^D = \Gamma_1 \cup \Gamma_{10}$, $\Gamma^R = \bigcup_{i=20}^{23} \Gamma_i$, and $a^R = \alpha \delta_{\Gamma_{22} \cup \Gamma_{23}}$.

We give in Listing 2.4 the corresponding Matlab code.

Listing 2.4: 2D Laplacian eigenvalues problem with mixed boundary conditions on a domain with 5 holes

```
% Setting mesh:
fullgeofile=fc_vfemp1.addon.eigs.get_geo(2,2,'disk5holes.geo');
meshfile=fc_oogmsh.buildmesh2d(fullgeofile,N);
Th=fc_simesh.siMesh(meshfile);
Th.info('verbose',true)

% Setting eBVP:
Lop=fc_vfemp1.Loperator(2,2,{1,0;0,1},[],[],[]);
pde=fc_vfemp1.PDE(Lop);
bvp=fc_vfemp1.BVP(Th,pde); % By default sets homogeneous Neumann on all boundaries
for lab=[1,10], bvp.setDirichlet( lab, 0);end
```

```
for lab=[22,23], bvp.setRobin( lab, 0, a);end

% Solving eBVP:
[eVec,lambda]=fc_vfemp1.addon.eigs.solve(bvp,'neigs',neigs,'sigma',sigma);

% Plotting eigenfunctions:
[eVec,lambda]=fc_vfemp1.addon.eigs.splitsort(bvp,eVec,lambda);
fc_vfemp1.addon.eigs.plot(Th,eVec,lambda);
```

We represent in Figure 2.10 the twenty four first eigenfunctions with $\mathrm{n_q} = 73620$, $\mathrm{n_{me}} = 145636$ and $\alpha = 10.$.

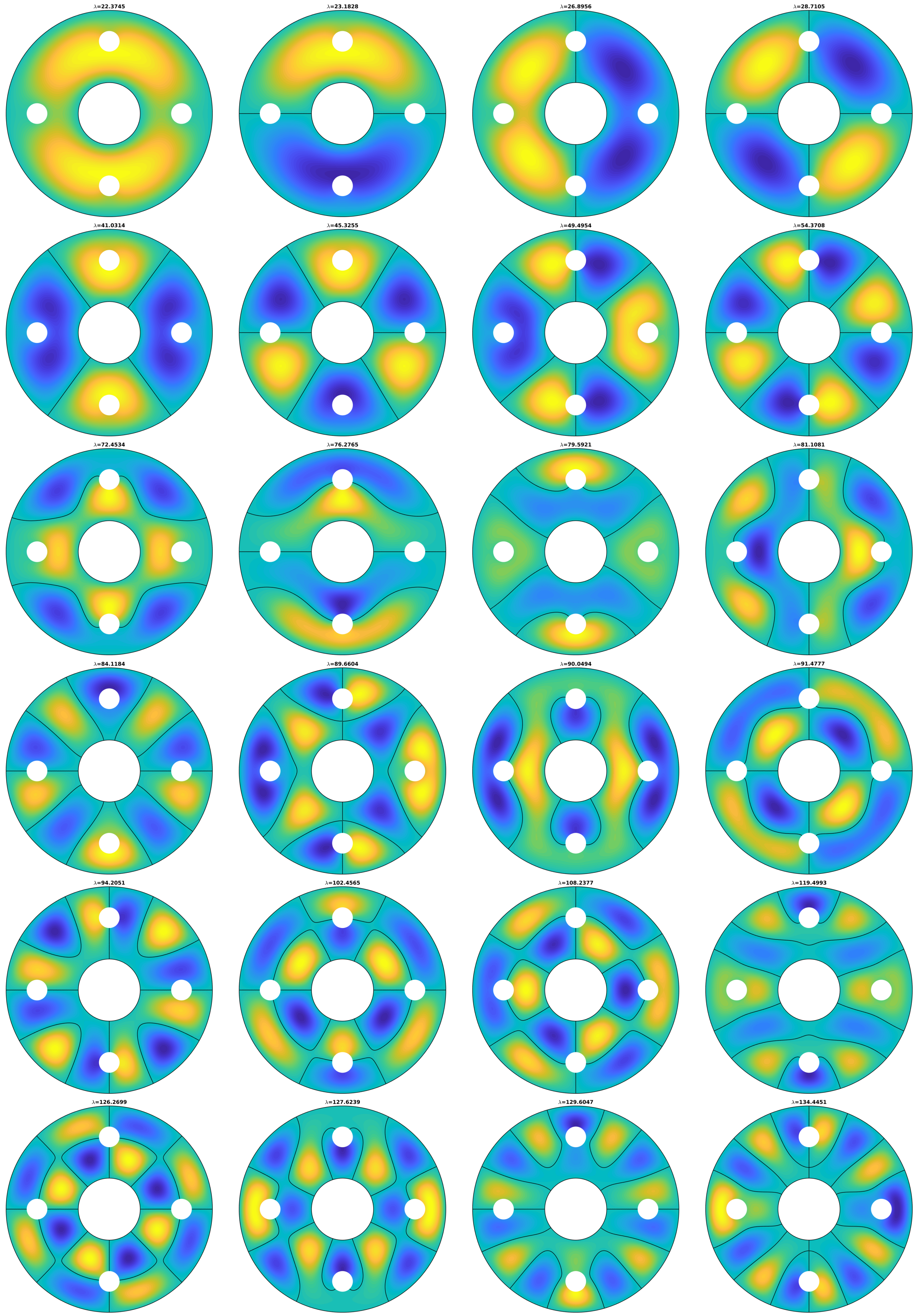


Figure 2.10: 2D Laplace with mixed boundary conditions : eigenfunctions of the smallest magnitude eigenvalues

2.2.3 Other 2D eigenvalues problems with Dirichlet boundary condition

Advection-Diffusion on the L-shaped domain.

We want to solve the eigenvalue problem given by

Usual EBVP 3 : 2D Advection-Diffusion eigenvalues problem with Dirichlet boundary condition

Find $(\lambda, u) \in \mathbb{K} \times \mathrm{H}^2(\Omega)$ such that

$$\begin{aligned} -\Delta u + \boldsymbol{\beta} . \boldsymbol{\nabla} u &= \lambda u && \text{in } \Omega, \\ u &= 0 && \text{on } \Gamma. \end{aligned}$$

with constant convection parameter $\boldsymbol{\beta} \in \mathbb{R}^2$.

From [4] section 6.52 page 122 the eigenvalues of *Usual* EBVP 3 are $\lambda_i^{\boldsymbol{\beta}} = |\boldsymbol{\beta}|/4 + \lambda_i$ where λ_i are the eigenvalues of the L-shaped Laplacian problem with Dirichlet boundary condition (the ten first are given in Table ??). We have for example

$$\begin{aligned} \lambda_1^{\boldsymbol{\beta}} &\approx |\boldsymbol{\beta}|/4 + 9.63972, & \lambda_3^{\boldsymbol{\beta}} &= |\boldsymbol{\beta}|/4 + 2\pi^2 \\ \lambda_5^{\boldsymbol{\beta}} &\approx |\boldsymbol{\beta}|/4 + 31.912636, & \lambda_8^{\boldsymbol{\beta}} = \lambda_9^{\boldsymbol{\beta}} &= |\boldsymbol{\beta}|/4 + 5\pi^2 \\ \lambda_{20}^{\boldsymbol{\beta}} &\approx |\boldsymbol{\beta}|/4 + 101.60529, & \lambda_{50}^{\boldsymbol{\beta}} &\approx |\boldsymbol{\beta}|/4 + 250.78548. \end{aligned}$$

We give in Listing 2.5 the corresponding Matlab code.

Listing 2.5: 2D L-shaped Advection-Diffusion problem with $\boldsymbol{\beta} = (3, 0)$: Matlab code

```matlab
% Setting mesh:
fullgeofile=fc_vfemp1.addon.eigs.get_geo(2,2,'Lshape');
meshfile=fc_oogmsh.buildmesh2d(fullgeofile,N);
Th=fc_simesh.siMesh(meshfile);
Th.info('verbose',true)

% Setting eBVP:
Lop=fc_vfemp1.Loperator(2,2,{1,0;0,1},[],b,[]);
pde=fc_vfemp1.PDE(Lop);
bvp=fc_vfemp1.BVP(Th,pde); % By default sets homogeneous Neumann on all boundaries
for lab=Th.get_labels(1), bvp.setDirichlet( lab, 0);end

% Solving eBVP:
[eVec,lambda]=fc_vfemp1.addon.eigs.solve(bvp,'neigs',neigs,'sigma',sigma);

% Plotting eigenfunctions:
[eVec,lambda]=fc_vfemp1.addon.eigs.splitsort(bvp,eVec,lambda);
fc_vfemp1.addon.eigs.plot(Th,eVec,lambda);
```

We give the computed values from a L-shaped mesh with $\mathrm{n_q} = 78980$, $\mathrm{n_{me}} = 156758$ and $h \approx 0.00862$. We represent in Figure 2.11 the twenty first eigenfunctions with $\boldsymbol{\beta} = (3, 0)$.

2.3 3D examples

2.3.1 Eigenvalues of the laplacian on the unit sphere with Dirichlet boundary condition

Application on the unit sphere. Let $\Omega \subset \mathbb{R}^3$ be the unit sphere meshed by gmsh and given in Figure 2.12.

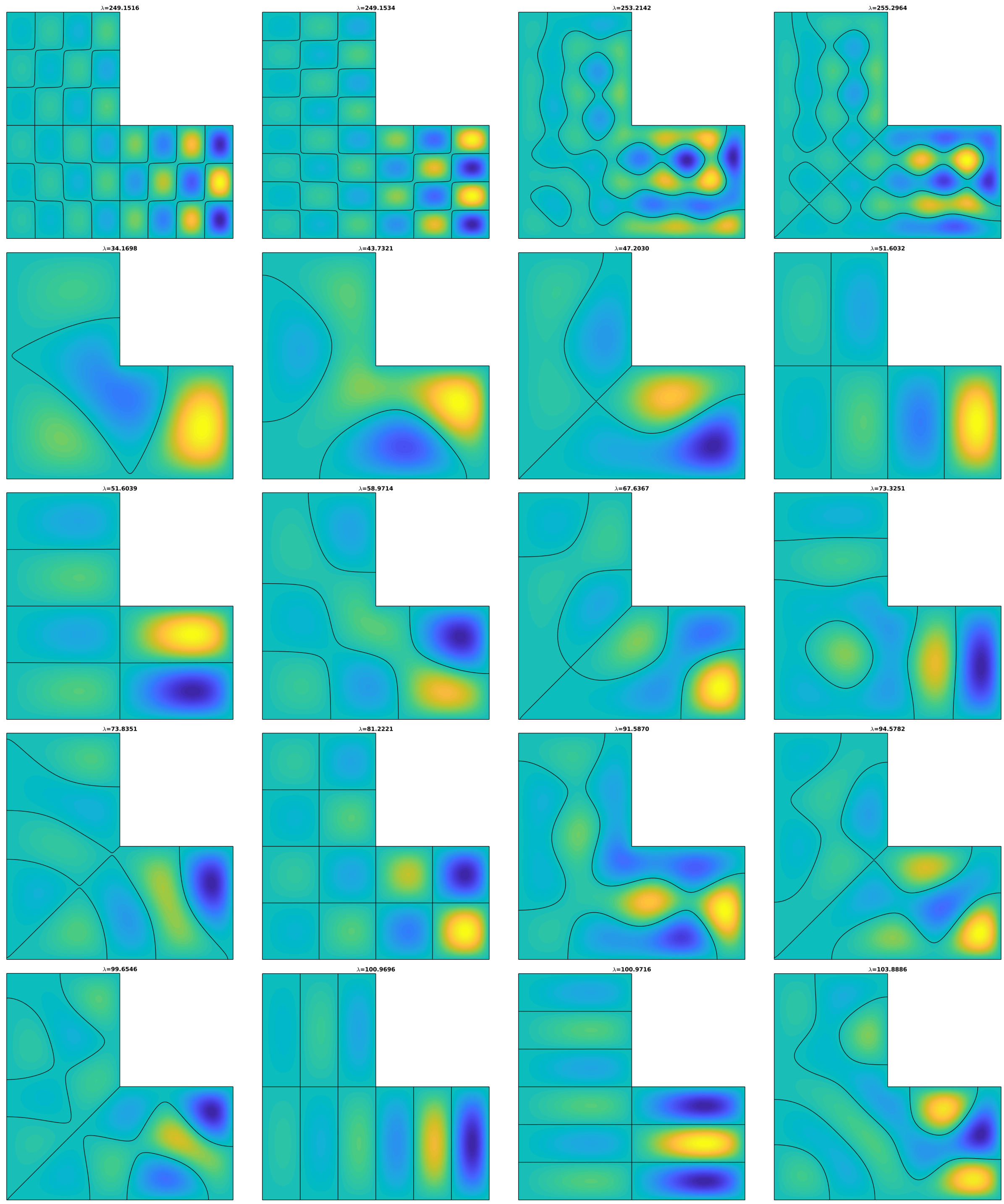


Figure 2.11: 2D L-shaped Advection-Diffusion problem with $\boldsymbol{\beta} = (3, 0)$. Eigenfunctions of the smallest magnitude eigenvalues ($\mathrm{n_q} = 89780$)

n	bounds of $\lambda_n^{\boldsymbol{\beta}}$ from [8]	computed here
1	$\lvert\boldsymbol{\beta}\rvert/4 + 9.63972384^{44}_{04} \approx 11.88972384$	11.891869
2	$\lvert\boldsymbol{\beta}\rvert/4 + 15.19725192^{66}_{59} \approx 17.44725192$	17.447524
3	$\lvert\boldsymbol{\beta}\rvert/4 + 2\pi^2 = 21.989208802178716$	21.989808
4	$\lvert\boldsymbol{\beta}\rvert/4 + 29.52148111^{42}_{38} \approx 31.77148111$	31.773171
5	$\lvert\boldsymbol{\beta}\rvert/4 + 31.9126359^{59}_{37} \approx 34.1626359$	34.169836
6	$\lvert\boldsymbol{\beta}\rvert/4 + 41.4745098^{92}_{66} \approx 43.7245099$	43.732143
7	$\lvert\boldsymbol{\beta}\rvert/4 + 44.9484877^{82}_{77} \approx 47.19848777$	47.202972
8	$\lvert\boldsymbol{\beta}\rvert/4 + 5\pi^2 = 51.598022005446794$	51.603181
9	$\lvert\boldsymbol{\beta}\rvert/4 + 5\pi^2 = 51.598022005446794$	51.603923
10	$\lvert\boldsymbol{\beta}\rvert/4 + 56.7096098^{90}_{18} \approx 58.9596098$	58.971443
20	$\lvert\boldsymbol{\beta}\rvert/4 + 101.60529 \approx 103.85529$	103.88860
50	$\lvert\boldsymbol{\beta}\rvert/4 + 250.78548 \approx 253.03548$	253.21421

Table 2.6: Eigenvalues of the L-shaped Advection-Diffusion problem with $\boldsymbol{\beta} = (3, 0)$.

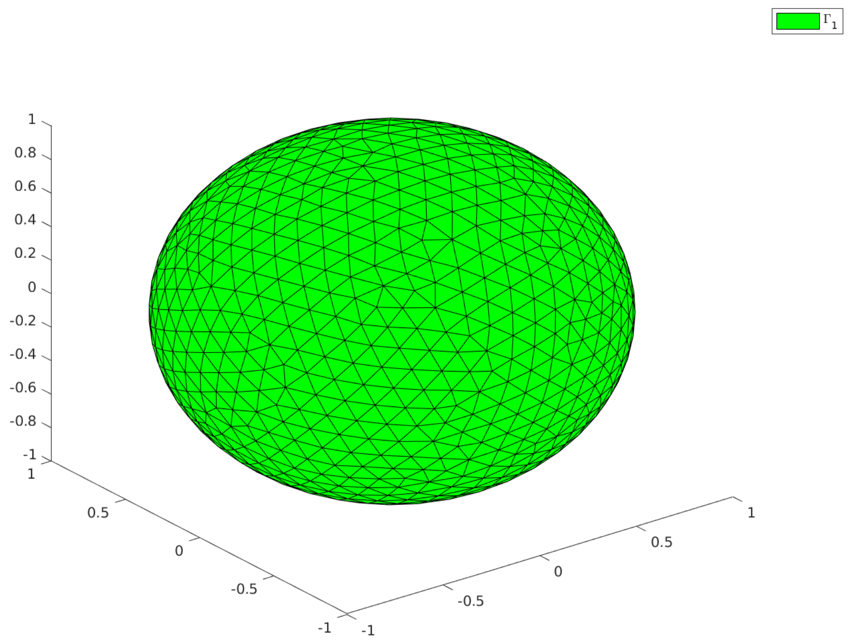


Figure 2.12: Boundary of the unit sphere

Part of the source code To compute the eigenvalues and the eigenfunctions of the Laplacian with Dirichlet boundary condition using $\mathbb{P}_1$-Lagrange finite elements one can used the Matlab command fc_vfemp1.addon.eigs.demos.dim3d.Laplacian02. Part of the source code (file `+fc_vfemp1/+addon/+eigs/+demos/+dim3d` is given in Listing 2.6

```
% Setting mesh:
fullgeofile=fc_vfemp1.addon.eigs.get_geo(3,3,'sphere.geo');
meshfile=fc_oogmsh.buildmesh3d(fullgeofile,N);
Th=fc_simesh.siMesh(meshfile);
Th.info('verbose',true)

% Setting eBVP:
Lop=fc_vfemp1.Loperator(3,3,{1,0,0;0,1,0;0,0,1},[],[],[]);
pde=fc_vfemp1.PDE(Lop);
bvp=fc_vfemp1.BVP(Th,pde);
bvp.setDirichlet(1, 0);
```

```
% Solving eBVP:
[eVec,lambda]=fc_vfemp1.addon.eigs.solve(bvp,'neigs',neigs,'sigma',sigma);

% Plotting eigenfunctions:
```

Listing 2.6: 3D Laplace eigenvalues problem with Dirichlet boundary condition on *sphere* domain

From [5], page 609, we get the eigenvalues of the laplacian on the sphere with Dirichlet boundary conditions. Let α_{nk} be the k-th zero of the Bessel function of the first kind $J_{n+1/2}$. The eigenvalues are given by

$$\lambda_{n,k} = \alpha_{nk}^2 \quad \forall n \in \mathbb{N}, \quad \forall k \in \mathbb{N}^*$$

In Table 2.7, the values of α_{nk} are given for $(n,k) \in [\![0,4]\!][\![1,5]\!]$. The eigenvalues have the degeneracy $2n+1$.

l	$J_{1/2}$	$J_{1+1/2}$	$J_{2+1/2}$	$J_{3+1/2}$	$J_{4+1/2}$	$J_{5+1/2}$	$J_{6+1/2}$
1	3.1415927	4.4934095	5.7634592	6.9879320	8.1825615	9.3558121	10.512835
2	6.2831853	7.7252518	9.0950113	10.417119	11.704907	12.966530	14.207392
3	9.4247780	10.904122	12.322941	13.698023	15.039665	16.354710	17.647975
4	12.566371	14.066194	15.514603	16.923621	18.301256	19.653152	20.983463
5	15.707963	17.220755	18.689036	20.121806	21.525418	22.904551	24.262768
6	18.849556	20.371303	21.853874	23.304247	24.727566	26.127750	27.507868

Table 2.7: Zeros of the Bessel function of the first kind $J_{n+1/2}$

	exact	**numerical**	**rel. error**
λ_1	9.8696044	9.8843887	1.498e-03
λ_2	20.190729	20.268363	3.845e-03
λ_3	20.190729	20.268576	3.856e-03
λ_4	20.190729	20.268690	3.861e-03
λ_5	33.217462	33.444466	6.834e-03
λ_6	33.217462	33.444913	6.847e-03
λ_7	33.217462	33.445149	6.854e-03
λ_8	33.217462	33.445816	6.875e-03
λ_9	33.217462	33.446448	6.894e-03
λ_{10}	39.478418	39.798711	8.113e-03
λ_{11}	48.831194	49.339144	1.040e-02
λ_{12}	48.831194	49.341745	1.046e-02
λ_{13}	48.831194	49.342522	1.047e-02
λ_{14}	48.831194	49.344420	1.051e-02
λ_{15}	48.831194	49.344782	1.052e-02
λ_{16}	48.831194	49.346092	1.054e-02
λ_{17}	48.831194	49.347897	1.058e-02
λ_{18}	59.679516	60.434811	1.266e-02
λ_{19}	59.679516	60.440039	1.274e-02
λ_{20}	59.679516	60.443164	1.280e-02
λ_{21}	66.954312	67.933217	1.462e-02
λ_{22}	66.954312	67.935441	1.465e-02
λ_{23}	66.954312	67.937650	1.469e-02
λ_{24}	66.954312	67.941095	1.474e-02

Table 2.8: Comparison between the twenty four first exact eigenvalues and numerical computation on mesh with $\mathrm{n_q} = 45371$ and $\mathrm{n_{me}} = 256980$

We represent in Figure 2.13 the eigenfunctions associated to the first twenty four smallest magnitude eigenvalues.

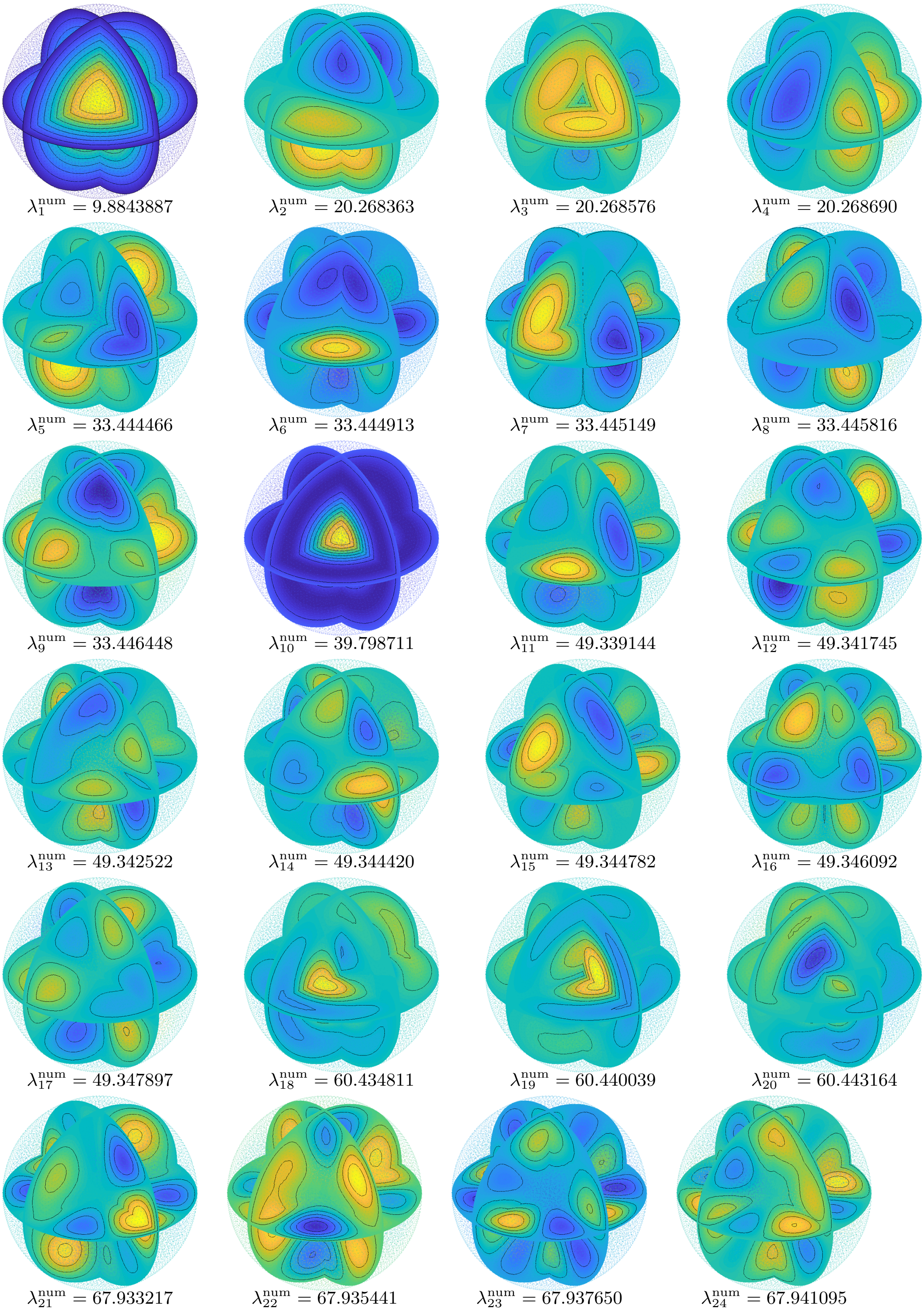


Figure 2.13: Dirichlet eigenvalue problem on the unit sphere : eigenfunctions associated to the first twenty four smallest magnitude eigenvalues.

Chapter 3

Generalized Eigenvalue vector BVP

The eigenvalue problems associated with *vector* BVP (1.14)-(1.16) can be written as

Vector EBVP 1 : generic problem

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

$$\mathcal{H}(\boldsymbol{u}) = \lambda\mathcal{B}(\boldsymbol{u}) \quad \text{in } \Omega, \tag{3.1}$$

$$\boldsymbol{u}_\alpha = 0 \quad \text{on } \Gamma_\alpha^D,\ \forall \alpha \in [\![1, m]\!], \tag{3.2}$$

$$\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}_\alpha}} + a_\alpha^R \boldsymbol{u}_\alpha = 0 \quad \text{on } \Gamma_\alpha^R,\ \forall \alpha \in [\![1, m]\!], \tag{3.3}$$

where $\mathcal{B}$ is a given $\mathcal{H}$-operator.

In most cases $\mathcal{B}$ is the identity operator ($\mathcal{B}$ is a diagonal $\mathcal{H}$-operator with $\mathcal{B}_{\alpha,\alpha} = \mathcal{L}_{\mathbb{O}_{d\times d}, \mathbf{0}_d, \mathbf{0}_d, 1}$, $\forall \alpha \in [\![1, m]\!]$).

3.1 Linear elasticity

3.1.1 Elasticity problem

Let $d = 2$ or $d = 3$. We consider here Hooke's law in linear elasticity, under small strain hypothesis (see for example [2]).

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

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

material	E $[psi](kg/s^2/mm)$	ρ in $[psi]kg/mm^3$	ν
Aluminium 6061-T6	68.26×10^6	2.7126×10^{-6}	0.33
Silicone Rubber	50×10^6	1.1×10^{-6}	0.33
Wood	23×10^6	0.95×10^{-6}	0.27
Nickel Aluminum Bronze 632	120.66×10^6	7.5843×10^{-6}	0.27

Table 3.1: Mechanical properties of some materials with GPa $= 10^9$Pa $= 10^6$kg/s^2/mm

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 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

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

For rubber material one can take $E = 21\,.10^5$Pa and $\nu = 0.48$.

The problem to solve is the following

Usual EBVP 4 : Elasticity problem

Find $(k, \boldsymbol{u}) = \mathbb{K} \times \mathrm{H}^2(\Omega)^d$ such that

$$\begin{aligned} -\operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u})) &= k\boldsymbol{u}, \quad \text{in } \Omega \subset \mathbb{R}^d, && (3.4) \\ \boldsymbol{\sigma}(\boldsymbol{u}).\boldsymbol{n} &= \boldsymbol{0} \quad \text{on } \Gamma^R, && (3.5) \\ \boldsymbol{u} &= \boldsymbol{0} \quad \text{on } \Gamma^D. && (3.6) \end{aligned}$$

We recall the following lemma (see [1])

Lemme 3.1

Let $\mathcal{H}^{\boldsymbol{\sigma}}$ be the $\mathcal{H}$-operator defined in (1.10) by

$$\mathcal{H}_{\alpha,\beta} = \mathcal{L}_{\mathbb{A}^{\alpha,\beta},\boldsymbol{0},\boldsymbol{0},0}, \quad \forall (\alpha,\beta) \in [\![1,d]\!]^2 \tag{3.7}$$

with

$$(\mathbb{A}^{\alpha,\beta})_{k,l} = \mu\delta_{\alpha\beta}\delta_{kl} + \mu\delta_{k\beta}\delta_{l\alpha} + \lambda\delta_{k\alpha}\delta_{l\beta}, \ \forall (k,l) \in [\![1,d]\!]^2. \tag{3.8}$$

Then, we have

$$\mathcal{H}^{\boldsymbol{\sigma}}(\boldsymbol{u}) = -\operatorname{div}\boldsymbol{\sigma}(\boldsymbol{u}) \tag{3.9}$$

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

$$\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}^{\boldsymbol{\sigma}}_\alpha}} = (\boldsymbol{\sigma}(\boldsymbol{u}).\boldsymbol{n})_\alpha. \tag{3.10}$$

The matrices $\mathbb{A}^{\alpha,\beta}$ of previous lemma are explicitely given by

- 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 $d = 3$,

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

So the elasticity problem (3.4) to (3.14) can be equivalently written as :

Vector EBVP 2 : Linear elasticity in dimension $d = 2$ or $d = 3$

Find $(k, \boldsymbol{u}) \in \mathbb{K} \times (\mathrm{H}^2(\Omega))^d$ such that

$$\mathcal{H}^{\boldsymbol{\sigma}}(\boldsymbol{u}) = k\mathcal{B}^{\boldsymbol{\sigma}}(\boldsymbol{u}), \quad \text{in } \Omega, \tag{3.11}$$
$$\frac{\partial \boldsymbol{u}}{\partial n_{\mathcal{H}^{\boldsymbol{\sigma}}_\alpha}} = 0, \quad \text{on } \Gamma^R_\alpha = \Gamma^R, \ \forall \alpha \in [\![1, d]\!] \tag{3.12}$$
$$\boldsymbol{u}_\alpha = 0, \quad \text{on } \Gamma^D_\alpha = \Gamma^D, \ \forall \alpha \in [\![1, d]\!]. \tag{3.13}$$

with $\mathcal{B}^{\boldsymbol{\sigma}}_{\alpha,\beta} = \delta_{\alpha,\beta}\mathcal{L}_{\mathbb{O},\mathbf{0},\mathbf{0},1}$.

3.1.2 2D tuning fork

2D tuning fork Let $\Omega \subset \mathbb{R}^2$ be a 2d tuning fork obtained from file `tuning-fork2D02.geo` by using **`gmsh`**(unit is millimeter). One can easily modify the 2d tuning fork parameters a, b, l, R and L (see Figure 3.1) when we generate the mesh:

```
fullgeofile=fc_vfemp1.addon.eigs.get_geo(2,2,'tuning fork2D02');
Lopts={'smooth␣4','setnumber␣L␣80','setnumber␣l␣60','setnumber␣R␣15', ...
       'setnumber␣a␣5','setnumber␣b␣8'};
meshfile=fc_oogmsh.gmsh.buildmesh2d(fullgeofile,60, ...
    'Loptions',Lopts,'force',true);
Th=fc_simesh.siMesh(meshfile);
```

Dirichlet on Γ_1 and Neumann on $\Gamma_2 \cup \Gamma_3$

The eigenvalue problem to solve is the following

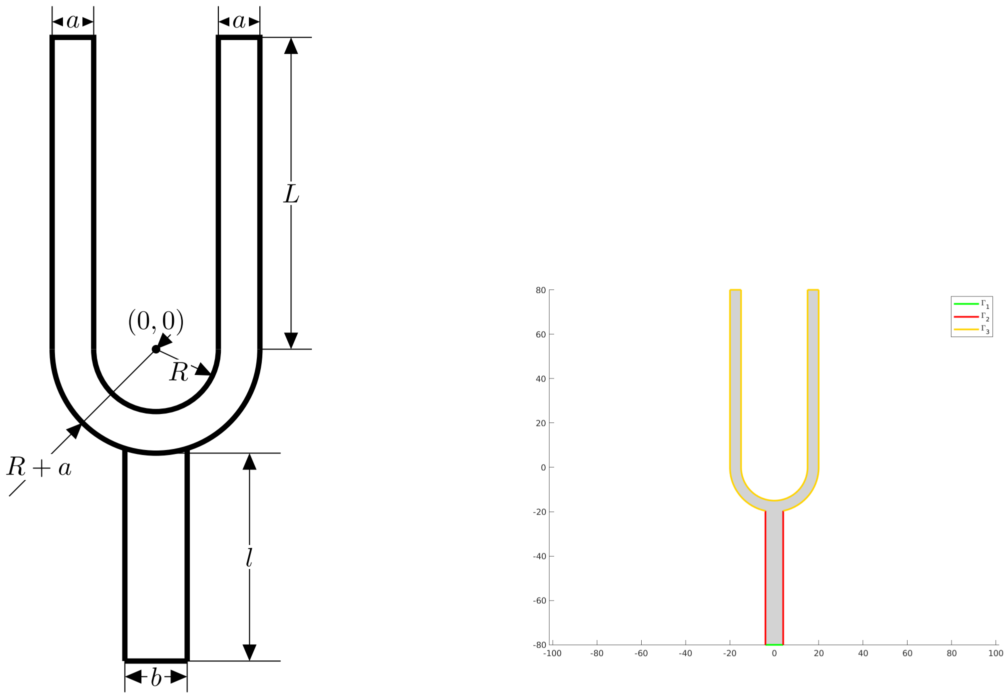


Figure 3.1: 2d tuning fork geometrical parameters (left) and its boundaries (right)

Vector EBVP 3 : 2D tuning fork with Dirichlet boundary condition on Γ_1

Find $(\kappa, \boldsymbol{u}) = \mathbb{K} \times \mathrm{H}^1(\Omega)^2$ such that

$$\begin{aligned} -\operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u})) &= \kappa\rho\boldsymbol{u}, \quad \text{in } \Omega \\ \boldsymbol{\sigma}(\boldsymbol{u}).\boldsymbol{n} &= \mathbf{0} \quad \text{on } \Gamma_2 \cup \Gamma_3, \\ \boldsymbol{u} &= \mathbf{0} \quad \text{on } \Gamma_1. \end{aligned}$$

The isotropic material is made of aluminium where its Poisson's ratio ν is 0.33, its Young's modulus E is 6.83×10^7 kg.s^{-2}.mm^{-1} and its density ρ is 2.71×10^{-6} kg.mm^{-3}.

We take $\Gamma^D = \Gamma_1$ and thus $\Gamma^R = \Gamma_2 \cup \Gamma_3$. For each eigenfunction $\boldsymbol{u}$, we represent $\sqrt{\boldsymbol{u}_1^2 + \boldsymbol{u}_2^2}$ in Figure 3.2 for the first twelve smallest magnitude eigenvalues. The mesh parameters are $\mathrm{n_q} = 41146$, $\mathrm{n_{me}} = 79533$ and $h = 0.26661$.

To compute the eigenvalues and the eigenfunctions of this problem by using $\mathbb{P}_1$-Lagrange finite elements one can used the Matlab script fc_vfemp1.addon.eigs.demos.dim2d.LinearElasticity02. Part of the source code is given in Listing 3.1

```
fprintf('  > Solving parameters: neigs=%d, sigma=%g, N=%d.\n',neigs,sigma,N);
fprintf('  > Material parameters: nu=%g, E=%g, rho=%g.\n',nu,E,rho);

% Computing the Lame parameters lambda and mu
mu= E/(2*(1+nu));
lambda = E*nu/((1+nu)*(1 2*nu));

% Setting mesh:
Lopts={' smooth 4',sprintf(' setnumber L %g',L), ...
          sprintf(' setnumber l %g',l), ...
          sprintf(' setnumber R %g',R), ...
          sprintf(' setnumber a %g',a), ...
          sprintf(' setnumber b %g',b) };
fullgeofile=fc_vfemp1.addon.eigs.get_geo(2,2,'tuning fork2D02');
meshfile=fc_oogmsh.buildmesh2d(fullgeofile,N,'Loptions',Lopts,'force',true);
```

```
Th=fc_simesh.siMesh(meshfile);
Th.info('verbose',true)

% Setting eBVP:
dim=2;d=2;m=2;
Hop=fc_vfemp1.operators.StiffElas(dim,lambda,mu);
pde=fc_vfemp1.PDE(Hop);
bvp=fc_vfemp1.BVP(Th,pde);
for lab=DirLabs, bvp.setDirichlet( lab, 0.,1:dim);end
Lop=fc_vfemp1.Loperator(dim,d,[],[],[],rho);
Bop=fc_vfemp1.Hoperator(dim,d,m);
for i=1:dim,Bop.H{i,i}=Lop;end

% Solving eBVP:
[eVec,eVal]=fc_vfemp1.addon.eigs.solve(bvp,'Bop',Bop,'neigs',neigs,'sigma',sigma);
```

Listing 3.1: 2D linear elasticity eigenvalues problem for the 2D tuning fork with a Dirichlet condition on Γ_1.

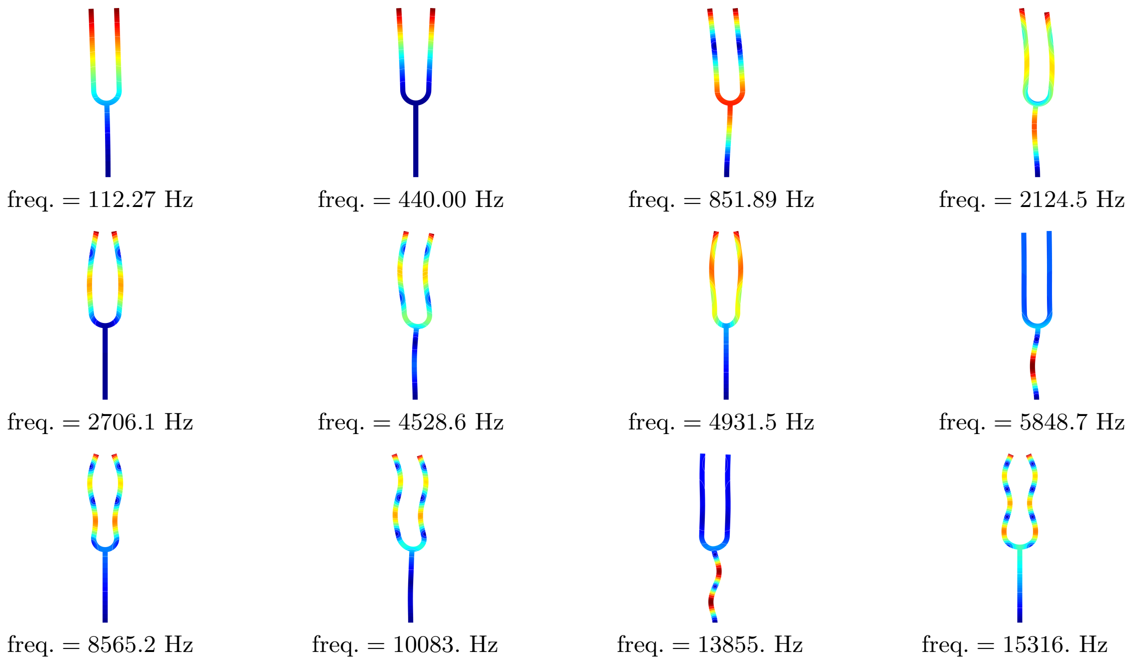


Figure 3.2: 2D linear elasticity for the tuning-fork with Dirichlet boundary condition on Γ_1 : eigenfunctions of the twelve smallest magnitude eigenvalues colorized with their euclidean norm. Eigenfunctions are streched to a maximum of 4.134 mm. The Tuning fork geometrical parameters in mm are $a = 5$, $b = 5$, $l = 70$, $R = 10$ and $L = 80.343$.

Neumann on all boundaries

The eigenvalue problem to solve is the following

Vector EBVP 4 : 2D tuning fork with Neumann boundary condition

Find $(\kappa, \boldsymbol{u}) = \mathbb{K} \times \mathrm{H}^1(\Omega)^2$ such that

$$\begin{aligned} -\operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u})) &= \kappa\rho\boldsymbol{u}, \quad \text{in } \Omega \\ \boldsymbol{\sigma}(\boldsymbol{u}).\boldsymbol{n} &= \boldsymbol{0} \quad \text{on } \Gamma_1 \cup \Gamma_2 \cup \Gamma_3, \end{aligned}$$

We take $\Gamma^D = \emptyset$ and thus $\Gamma^R = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$. For each eigenfunction $\boldsymbol{u}$, we represent $\sqrt{\boldsymbol{u}_1^2 + \boldsymbol{u}_2^2}$ in Figure 3.3 for the first twelve smallest magnitude eigenvalues. The mesh parameters are $\mathrm{n_q} = 41146$, $\mathrm{n_{me}} = 79533$ and $h =$ `2.666e-01`.

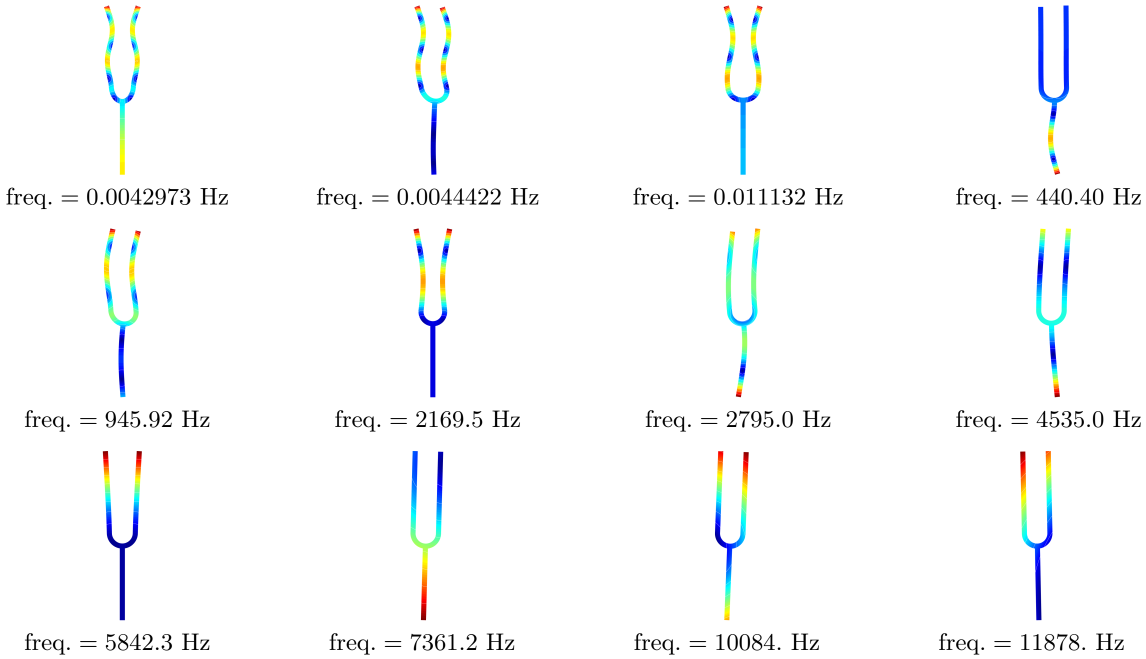


Figure 3.3: 2D linear elasticity for the tuning-fork with Neumann boundary conditions : eigenfunctions of the twelve smallest magnitude eigenvalues colorized with their euclidean norm. Eigenfunctions are streched to a maximum of 4.134 mm. The Tuning fork geometrical parameters in mm are $a = 5$, $b = 5$, $l = 70$, $R = 10$ and $L = 80.343$.

3.1.3 3D tuning fork

2D tuning fork Let $\Omega \subset \mathbb{R}^3$ be a 3d tuning fork composed of rods of diameter r. The geometry is described in Figure 3.4 and the meshes can be obtained by using `gmsh` (version $>=$ 3.0.0) with the file `tuning_fork_02.geo`.

The eigenvalue problem to solve is the following

Usual EBVP 5 : 3D tuning fork with Dirichlet boundary condition on Γ_1

Find $(\kappa, \boldsymbol{u}) = \mathbb{K} \times \mathrm{H}^1(\Omega)^3$ such that

$$\begin{aligned} -\operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u})) &= \kappa\rho\boldsymbol{u}, \quad \text{in } \Omega \\ \boldsymbol{\sigma}(\boldsymbol{u}).\boldsymbol{n} &= \boldsymbol{0} \quad \text{on } \Gamma_2 \cup \Gamma_3, \\ \boldsymbol{u} &= \boldsymbol{0} \quad \text{on } \Gamma_1. \end{aligned}$$

The geometrical parameters, in millimeters, are $L = 123.41$, $R = 40$, $l = 100$ and $r = 10$. The isotropic material is made of aluminium where its Poisson's ratio ν is 0.33, its Young's modulus E is 6.83×10^7 kg.s^{-2}.mm^{-1} and its density ρ is 2.71×10^{-6} kg.mm^{-3}.

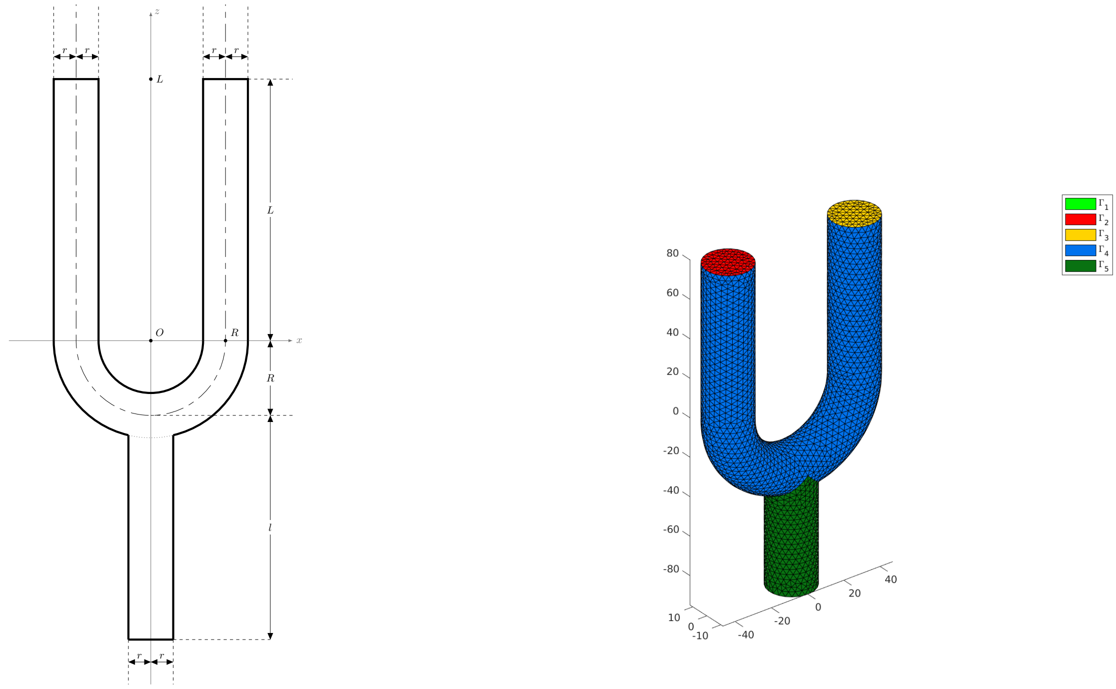


Figure 3.4: 3d tuning fork geometrical parameters (left) and its boundaries (right)

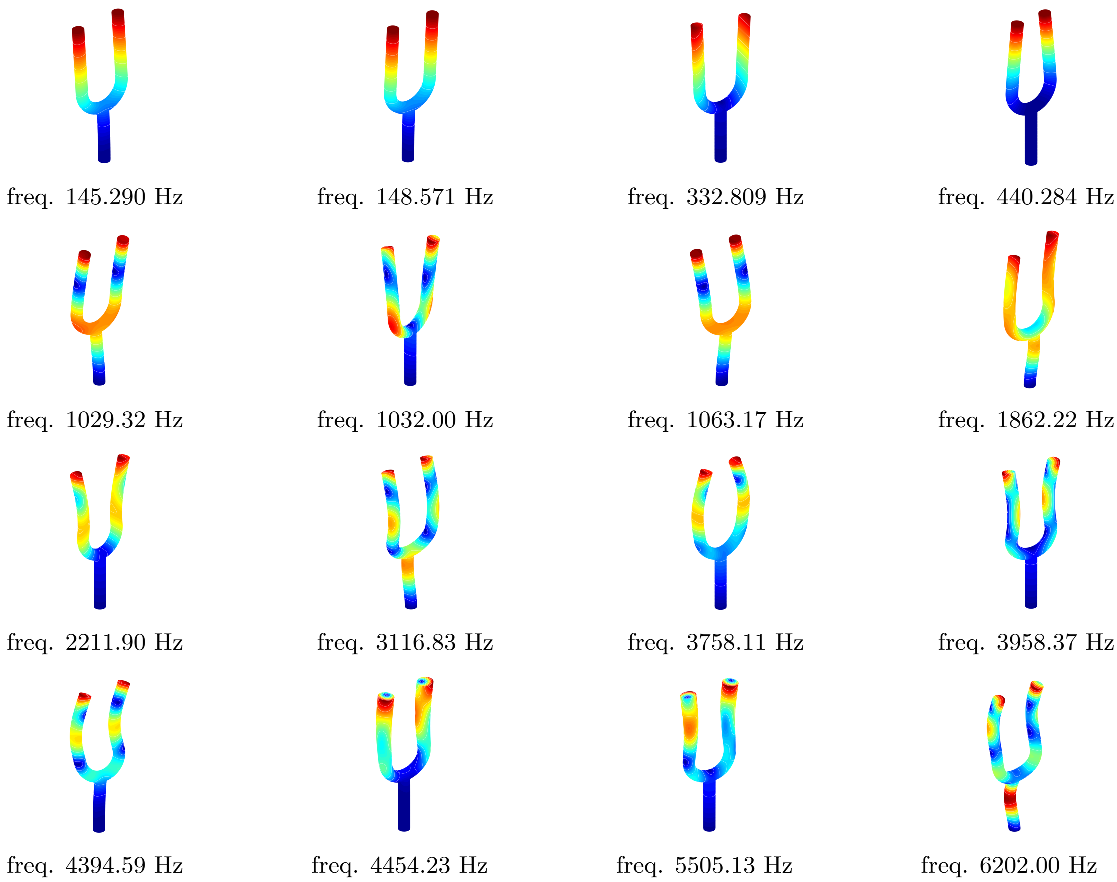


Figure 3.5: 3D Tuning fork with Neumann boundary conditions on $\Gamma_2 \cup \Gamma_3$ and Dirichlet on Γ_1: eigenfunctions of the smallest magnitude eigenvalues streched to a maximum of 13.17 mm.

3.2 Biharmonic eigenvalue BVP for plate vibration

The biharmonic eigenvalue problem for plate vibration is to find $u \neq 0$ and $\lambda \in \mathbb{R}$ such that

$$\Delta^2 u = \lambda u, \quad \text{in } \Omega \tag{3.14}$$

where boundary conditions on Γ can be

- *Clamped Plate* (CP) or *pure Dirichlet* type:

$$u = \frac{\partial u}{\partial \boldsymbol{n}} = 0 \tag{3.15}$$

- *Simply Supported Plate* (SSP) or *Navier* type :

$$u = \Delta u = 0 \tag{3.16}$$

- *Cahn-Hilliard* (CH) type

$$\frac{\partial u}{\partial n} = \frac{\partial \Delta u}{\partial n} = 0 \tag{3.17}$$

Classically the fourth-order PDE (3.14) is converted to the two second-order PDE

$$-\Delta v \quad = \quad \lambda u \tag{3.18}$$
$$-\Delta u \quad = \quad v \tag{3.19}$$

So with the the $\mathcal{H}$-operator $\mathcal{G}$ defined in (A.9) these two PDE can be written as

$$\mathcal{G}\begin{pmatrix} u \\ v \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} 0 & \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0} \\ \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0} & \mathcal{L}_{\mathbb{O},\mathbf{0},\mathbf{0},-1} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \lambda u \\ 0 \end{pmatrix} \tag{3.20}$$

If we define the $\mathcal{B}$ operator as

$$\mathcal{B} \stackrel{\text{def}}{=} \begin{pmatrix} \mathcal{L}_{\mathbb{O},\mathbf{0},\mathbf{0},1} & 0 \\ 0 & 0 \end{pmatrix} \tag{3.21}$$

the two PDEs (3.18) can be equivalently written as

$$\mathcal{G}\begin{pmatrix} u \\ v \end{pmatrix} = \lambda \mathcal{B} \begin{pmatrix} u \\ v \end{pmatrix} \tag{3.22}$$

The code to build the two operators $\mathcal{G}$ and $\mathcal{B}$ is given in Listing 3.2.

```
Gop=fc_vfemp1.Hoperator(2,2,2);
Lop=fc_vfemp1.Loperator(2,2,{1,[];[],1},[],[],[]);
Gop.set(1,2,Lop);
Gop.set(2,1,Lop);
Gop.set(2,2,fc_vfemp1.Loperator(2,2,[],[],[], 1));
Bop=fc_vfemp1.Hoperator(2,2,2);
Bop.set(1,1,fc_vfemp1.Loperator(2,2,[],[],[],1));
```

Listing 3.2: Setting the $\mathcal{G}$ and $\mathcal{B}$ operator.

3.2.1 Simply Supported Plate

eBVP 1 : biharmonic eigenvalue problem for plate vibration with simply supported plate boundary conditions

Find $\lambda \in \mathbb{K}$ and $u \neq 0$ such that

$$\Delta^2 u = \lambda u, \quad \text{in } \Omega$$
$$u = 0 \quad \text{and} \quad \Delta u = 0, \quad \text{on } \Gamma$$

This problem can be rewritten with operators $\mathcal{G}$ and $\mathcal{B}$ defined respectively in (3.20) and (3.21). Let $v = -\Delta u$ and $\boldsymbol{w} = (u, v)$, previous problem becomes

***Vector* EBVP 5 : biharmonic eigenvalue problem for plate vibration with simply supported plate boundary conditions with $\mathcal{H}$-operators**

Find $\lambda \in \mathbb{K}$ and $\boldsymbol{w} = (w_1, w_2) \in (\mathrm{H}^2(\Omega))^2$, $\boldsymbol{w} \neq 0$, such that

$$\begin{aligned} \mathcal{G}(\boldsymbol{w}) &= \lambda \mathcal{B}(\boldsymbol{w}) && \text{in } \Omega, \\ w_1 = 0 \quad \text{and} \quad w_2 &= 0 && \text{on } \Gamma. \end{aligned}$$

remark 3.2

On convex domains, the biharmonic eigenvalues are just the squares of the eigenvalues of the Laplace operator with the homogeneous Dirichlet boundary condition and associated eigenfunctions are the same.

Application on $\Omega = [0, L] \times [0, H]$

The eigenvalues and the eigenvectors of the Laplacian with Dirichlet boundary conditions are given in (2.6) and Table 2.2 for $L = 2$ and $H = 3$. In Listing 3.3, the part of the code which solve the biharmonic eigenvalue problem for plate vibration with simply supported plate boundary conditions is given.

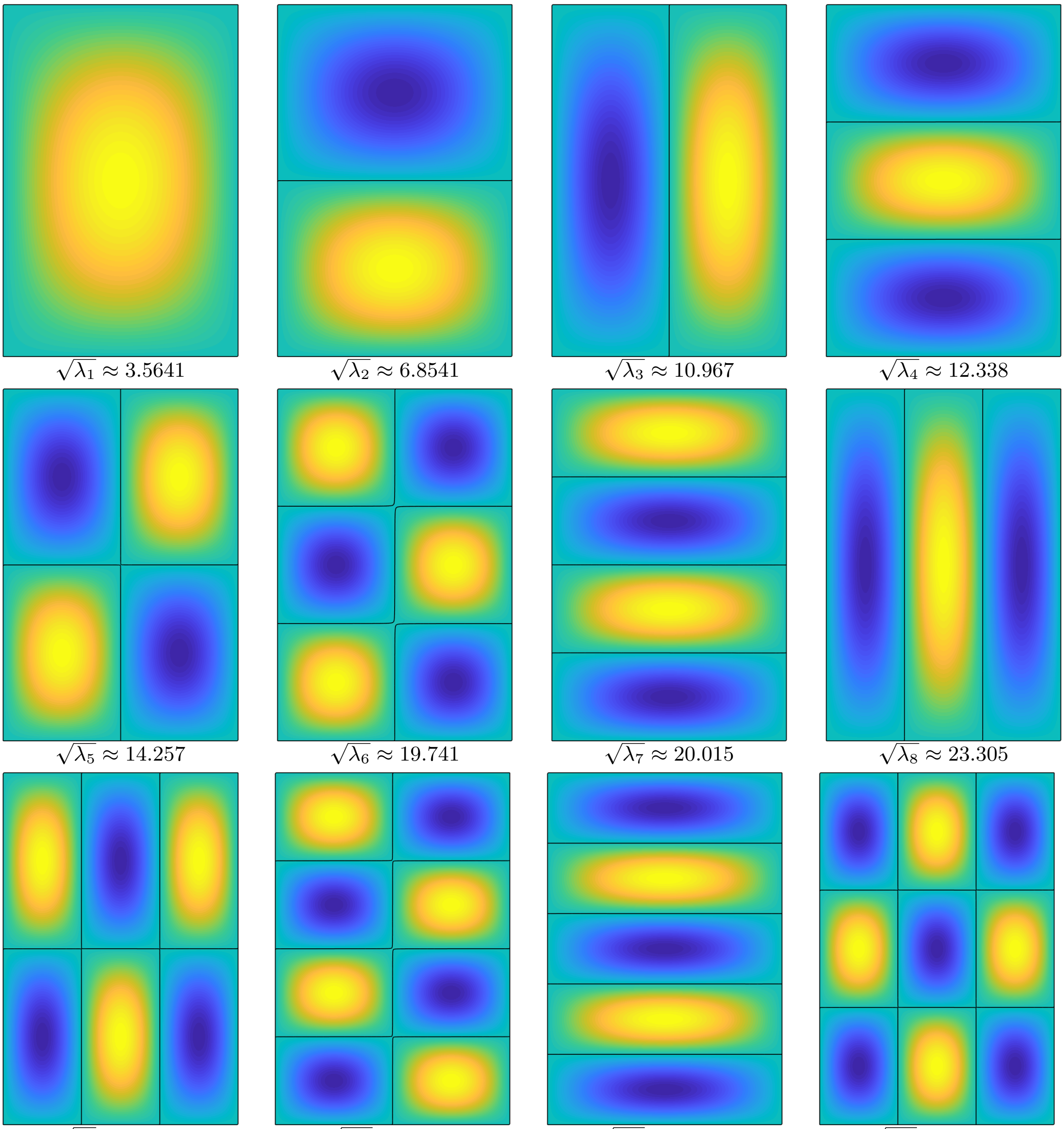
```
dim=2;d=2;m=2;
% Setting mesh:
Th=fc_simesh.hypercube(2,[L*N,H*N],'trans',@(q) [L*q(1,:);H*q(2,:)]);

% Setting eBVP:
Gop=fc_vfemp1.Hoperator(dim,d,m);
Lop=fc_vfemp1.Loperator(dim,d,{1,[];[],1},[],[],[]);
Gop.set(1,2,Lop);
Gop.set(2,1,Lop);
Gop.set(2,2,fc_vfemp1.Loperator(dim,d,[],[],[], 1));
Bop=fc_vfemp1.Hoperator(dim,d,m);
Bop.set(1,1,fc_vfemp1.Loperator(dim,d,[],[],[],1));
pde=fc_vfemp1.PDE(Gop);
bvp=fc_vfemp1.BVP(Th,pde);
for lab=Th.sThlab(Th.find(1)), bvp.setDirichlet( lab, 0.,1:2);end

% Solving eBVP:
```

Listing 3.3: biharmonic eigenvalue problem for plate vibration with simply supported plate boundary conditions $\Omega = [0, 123.41] \times [0, 3]$.

We represent in Figure 3.6 the twelve first eigenvectors obtained by using the command given in Listing 3.4.

```
neigs=12;N=150;sigma=0;L=2;H=3;isTitle=false;
fc_vfemp1.addon.eigs.demos.dim2d.Biharmonic01
```

Listing 3.4: Biharmonic eigenvalues B.V.P. with S.S.P. boundary conditions on the unit square



$\sqrt{\lambda_1} \approx 3.5641$ $\sqrt{\lambda_2} \approx 6.8541$ $\sqrt{\lambda_3} \approx 10.967$ $\sqrt{\lambda_4} \approx 12.338$

$\sqrt{\lambda_5} \approx 14.257$ $\sqrt{\lambda_6} \approx 19.741$ $\sqrt{\lambda_7} \approx 20.015$ $\sqrt{\lambda_8} \approx 23.305$

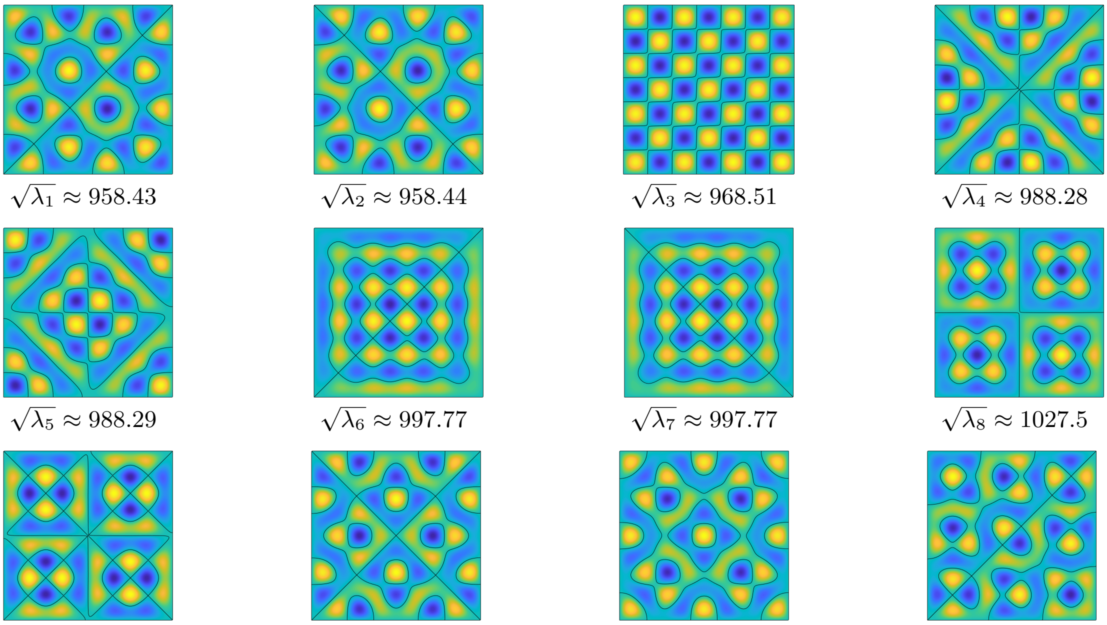
$\sqrt{\lambda_9} \approx 26.596$ $\sqrt{\lambda_{10}} \approx 27.420$ $\sqrt{\lambda_{11}} \approx 29.887$ $\sqrt{\lambda_{12}} \approx 32.082$

Figure 3.6: 2D Biharmonic eigenvalues B.V.P. with S.S.P. boundary conditions on the unit square: smallest eigenvalues with their eigenfunctions

We represent in Figure 3.7 the twelve eigenvectors associated with the eigenvalues closest to 1000^2 and obtained by using the command given in Listing 3.5.

```
[Th,eVec,lambda,info]= fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(300, ...
    'SSP',1:4,'L',1,'H',1,'sigma',1000^2,'neigs',12,'title',false);
```

Listing 3.5: Biharmonic eigenvalues B.V.P. with S.S.P. boundary conditions on the unit square



$\sqrt{\lambda_1} \approx 958.43$ $\sqrt{\lambda_2} \approx 958.44$ $\sqrt{\lambda_3} \approx 968.51$ $\sqrt{\lambda_4} \approx 988.28$

$\sqrt{\lambda_5} \approx 988.29$ $\sqrt{\lambda_6} \approx 997.77$ $\sqrt{\lambda_7} \approx 997.77$ $\sqrt{\lambda_8} \approx 1027.5$

$\sqrt{\lambda_9} \approx 1027.5$ $\sqrt{\lambda_{10}} \approx 1047.6$ $\sqrt{\lambda_{11}} \approx 1047.6$ $\sqrt{\lambda_{12}} \approx 1077.0$

Figure 3.7: 2D Biharmonic eigenvalues B.V.P. with S.S.P. boundary conditions on the unit square: eigenvalues closest to 1000^2 with their eigenfunctions

Application on a disk with 5 holes

We solve the eigenvalue problems eBVP 1 on the disk with five holes given in Figure 2.9 We represent in Figure 3.8 the twelve first eigenvectors obtained by using the command given in Listing 3.6.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'geofile','disk5holes','SSP',[1,10,20:23],'sigma',0,'neigs',12, ...
    'title',false);
```

Listing 3.6: Biharmonic eigenvalues B.V.P. with S.S.P. boundary conditions on a disk with five holes

We represent in Figure 3.9 the twelve eigenvectors associated with the eigenvalues closest to 1000^2 and obtained by using the command given in Listing 3.7.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'geofile','disk5holes','SSP',[1,10,20:23],'sigma',0,'neigs',12, ...
    'title',false);
```

Listing 3.7: Biharmonic eigenvalues B.V.P. with S.S.P. boundary conditions on a disk with five holes

3.2.2 Clamped Plate boundary condition

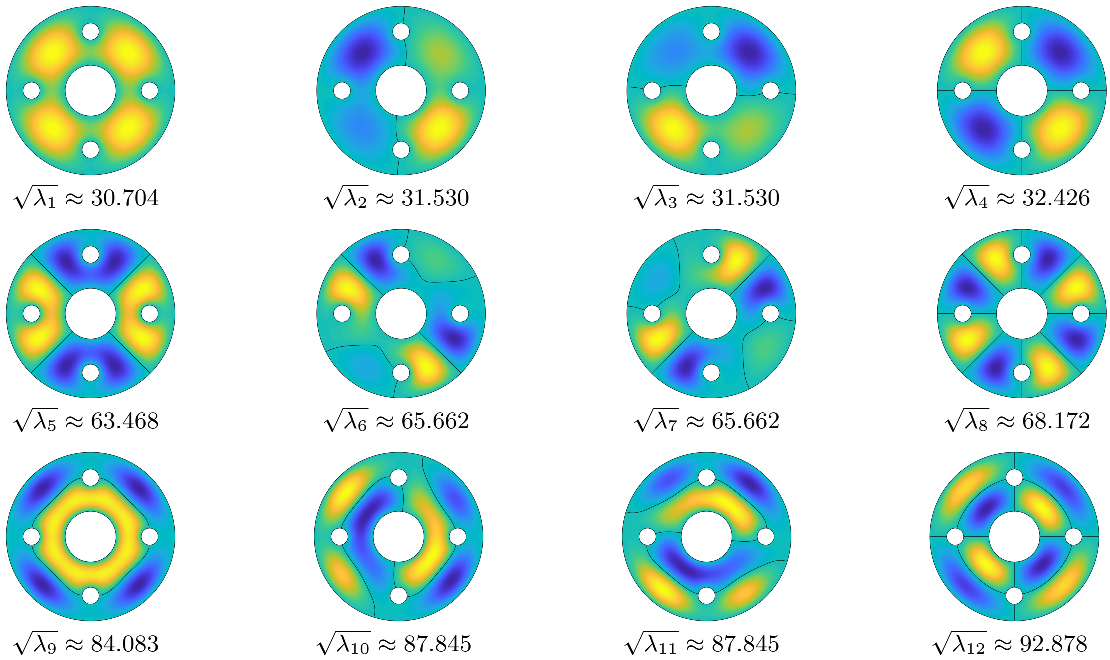


Figure 3.8: 2D Biharmonic eigenvalues B.V.P. with S.S.P. boundary conditions on a disk with five holes: smallest eigenvalues with their eigenfunctions

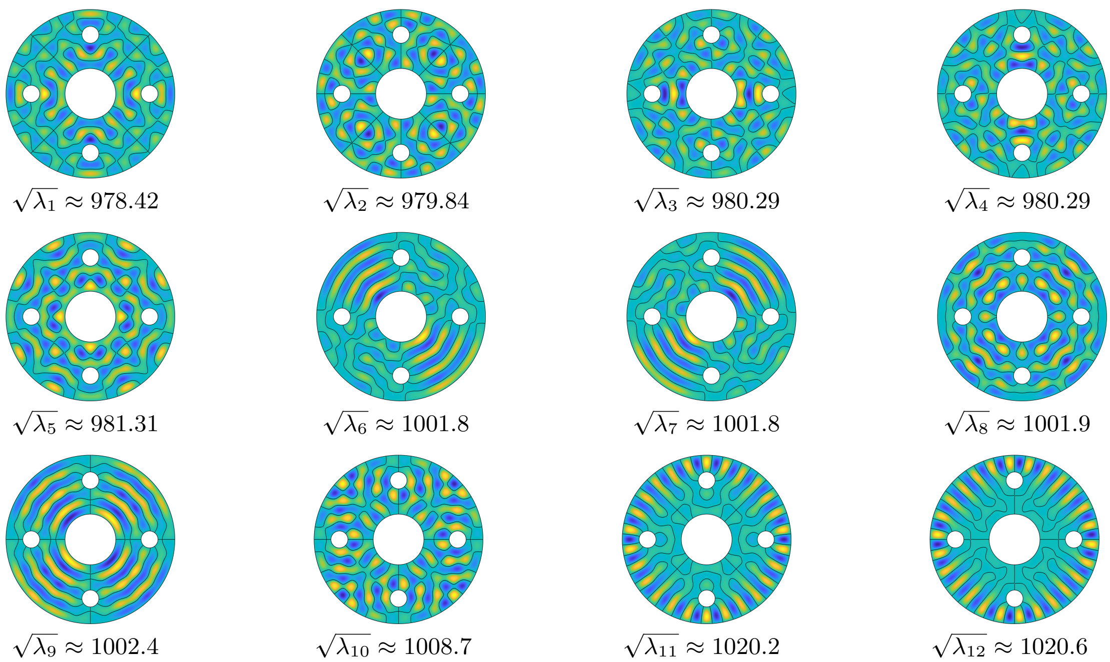


Figure 3.9: 2D Biharmonic eigenvalues B.V.P. with S.S.P. boundary conditions on a disk with five holes: eigenvalues closest to 1000^2 with their eigenfunctions

eBVP 2 : biharmonic eigenvalue problem for plate vibration with simply supported plate boundary conditions

Find $\lambda \in \mathbb{K}$ and $u \neq 0$ such that

$$\Delta^2 u = \lambda u, \quad \text{in } \Omega$$
$$u = 0 \quad \text{and} \quad fcfunpdiffv\boldsymbol{n} = 0, \quad \text{on } \Gamma$$

This problem can be rewritten with operators $\mathcal{G}$ and $\mathcal{B}$ defined respectively in (3.20) and (3.21). Let $v = -\Delta u$ and $\boldsymbol{w} = (u, v)$, previous problem becomes

***Vector* EBVP 6 : biharmonic eigenvalue problem for plate vibration with simply supported plate boundary conditions with $\mathcal{H}$-operators**

Find $\lambda \in \mathbb{K}$ and $\boldsymbol{w} = (w_1, w_2) \in (\mathrm{H}^2(\Omega))^2$, $\boldsymbol{w} \neq 0$, such that

$$\begin{aligned} \mathcal{G}(\boldsymbol{w}) &= \lambda \mathcal{B}(\boldsymbol{w}) && \text{in } \Omega, \\ w_1 = 0 \quad \text{and} \quad \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_2}} &= 0 && \text{on } \Gamma. \end{aligned}$$

Application on $\Omega = [0, L] \times [0, H]$

The eigenvalues and the eigenvectors of the Laplacian with Dirichlet boundary conditions are given in (2.6) and Table 2.1. In Listing 3.13, the part of the code which solve the biharmonic eigenvalue problem for plate vibration with simply supported plate boundary conditions is given.

```
pde=fc_vfemp1.PDE(Gop);
bvp=fc_vfemp1.BVP(Th,pde);
for lab=Th.get_labels(1)
  bvp.setDirichlet( lab, 0.,[],1);
  bvp.setRobin( lab, 0.,[],2);
end
[eVec,eVal,flag]=fc_vfemp1.addon.eigs.solve(bvp,'Bop',Bop,'neigs',12,'sigma',0);
```

Listing 3.8: Setting biharmonic eigenvalue problem for plate vibration with simply supported plate boundary conditions $\Omega = [0, 1] \times [0, 1]$. Using operators defined in Listing 3.2.

We represent in Figure 3.10 the twelve first eigenvectors obtained by using the command given in Listing 3.9.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'CP',1:4,'L',1,'H',1,'sigma',0,'neigs',12,'title',false);
```

Listing 3.9: Biharmonic eigenvalues B.V.P. with C.P. boundary conditions on the unit square

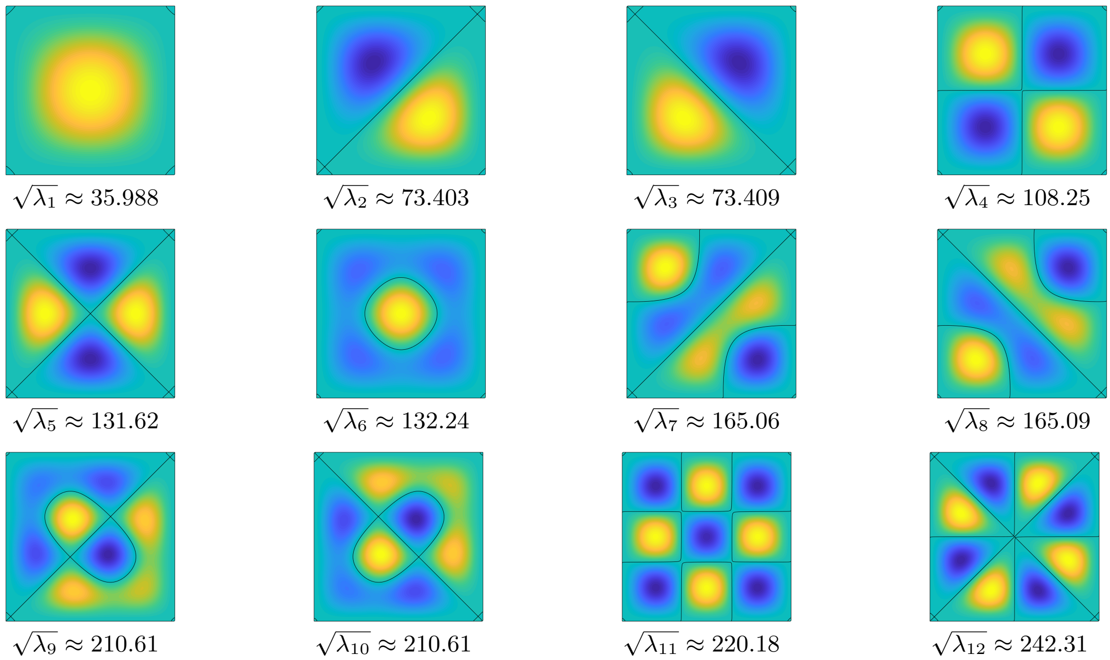


Figure 3.10: 2D Biharmonic eigenvalues B.V.P. with C.P. boundary conditions on the unit square: smallest eigenvalues with their eigenfunctions

We represent in Figure 3.11 the twelve eigenvectors associated with the eigenvalues closest to 1000^2 and obtained by using the command given in Listing 3.10.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'CP',1:4,'L',1,'H',1,'sigma',0,'neigs',12,'title',false);
```

Listing 3.10: Biharmonic eigenvalues B.V.P. with C.P. boundary conditions on the unit square

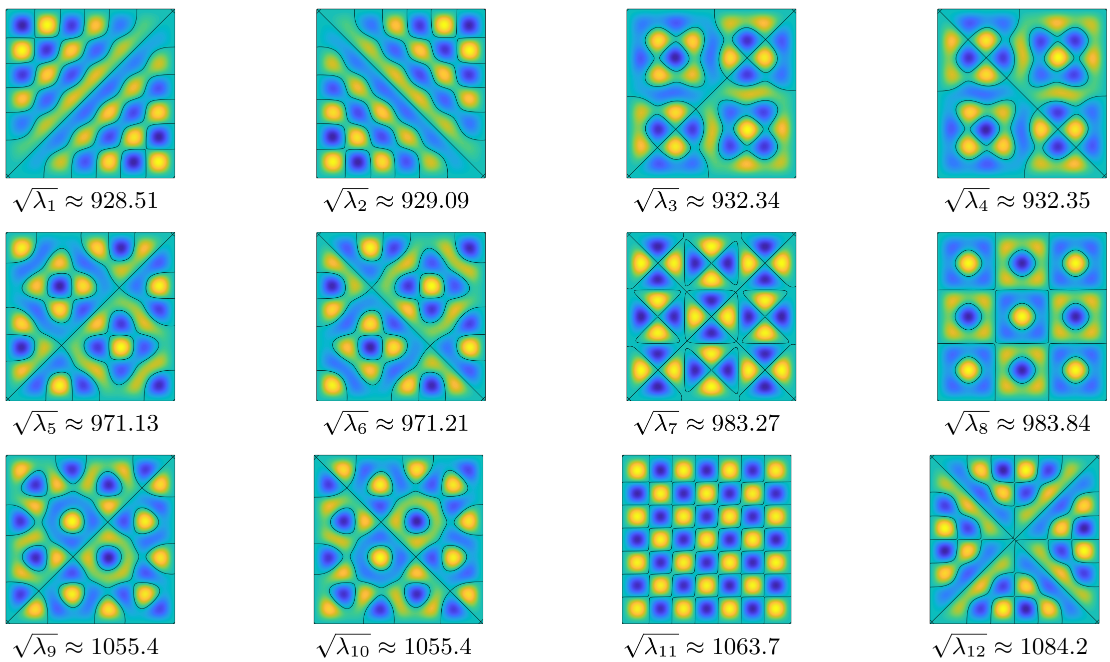


Figure 3.11: 2D Biharmonic eigenvalues B.V.P. with C.P. boundary conditions on the unit square: eigenvalues closest to 1000^2 with their eigenfunctions

Application on a disk with 5 holes

We solve the eigenvalue problems eBVP 1 on the disk with five holes given in Figure 2.9 We represent in Figure 3.12 the twelve first eigenvectors obtained by using the command given in Listing 3.11.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'geofile','disk5holes','CP',[1,10,20:23],'sigma',0,'neigs',12, 'title',false);
```

Listing 3.11: Biharmonic eigenvalues B.V.P. with C.P. boundary conditions on a disk with five holes

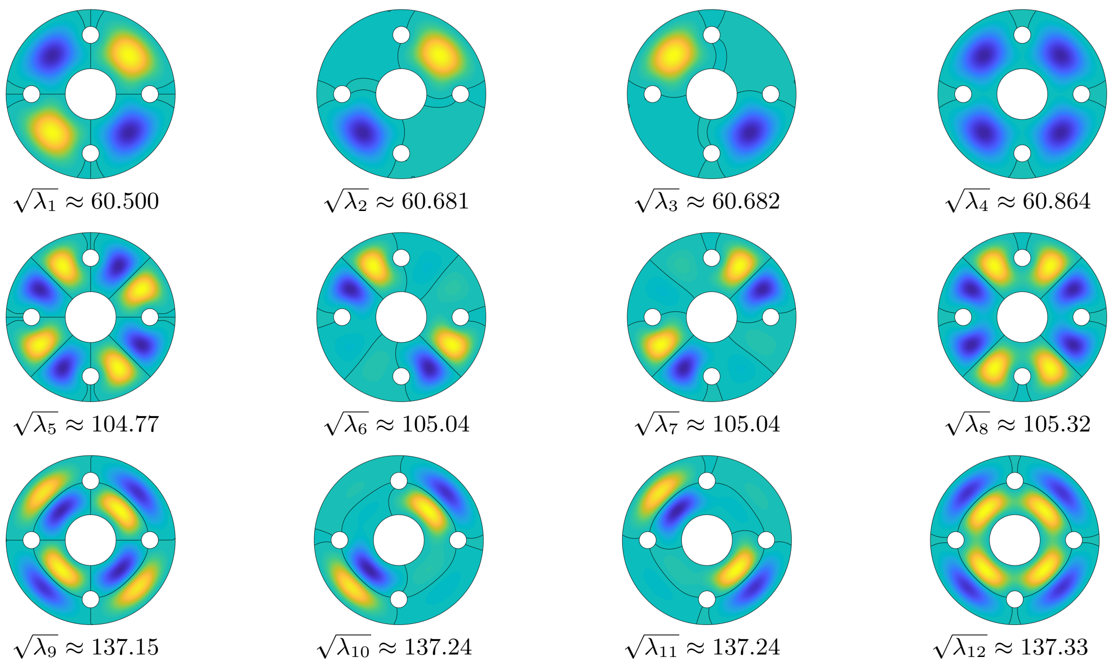
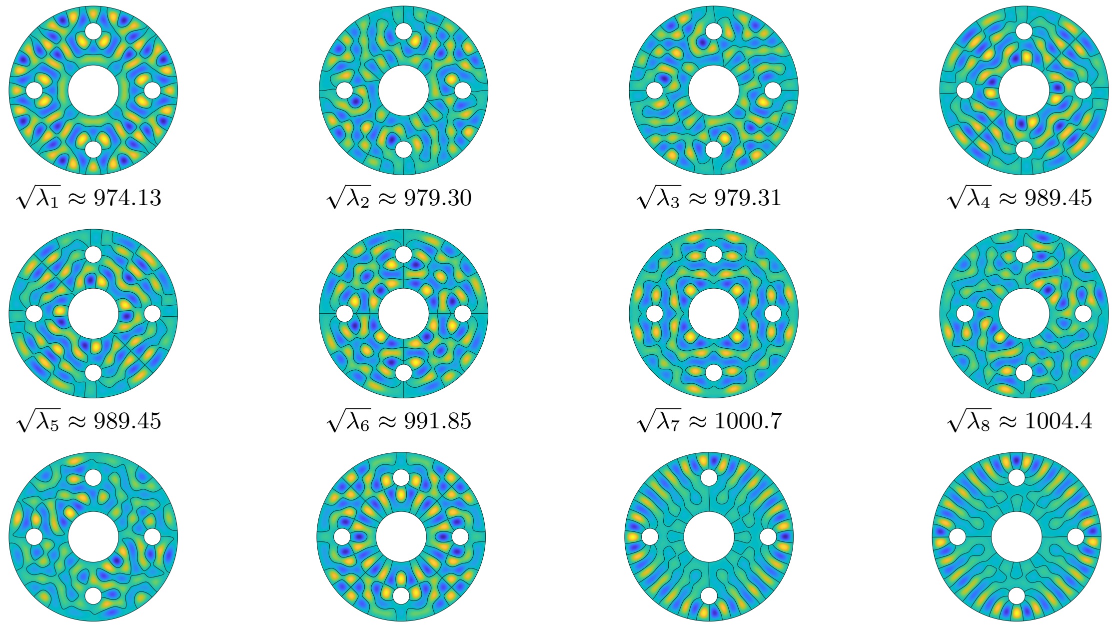


Figure 3.12: 2D Biharmonic eigenvalues B.V.P. with C.P. boundary conditions on a disk with five holes: smallest eigenvalues with their eigenfunctions

We represent in Figure 3.13 the twelve eigenvectors associated with the eigenvalues closest to 1000^2 and obtained by using the command given in Listing 3.12.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'geofile','disk5holes','CP',[1,10,20:23],'sigma',0,'neigs',12, 'title',false);
```

Listing 3.12: Biharmonic eigenvalues B.V.P. with C.P. boundary conditions on a disk with five holes



$\sqrt{\lambda_1} \approx 974.13$ $\sqrt{\lambda_2} \approx 979.30$ $\sqrt{\lambda_3} \approx 979.31$ $\sqrt{\lambda_4} \approx 989.45$

$\sqrt{\lambda_5} \approx 989.45$ $\sqrt{\lambda_6} \approx 991.85$ $\sqrt{\lambda_7} \approx 1000.7$ $\sqrt{\lambda_8} \approx 1004.4$

$\sqrt{\lambda_9} \approx 1004.4$ $\sqrt{\lambda_{10}} \approx 1012.5$ $\sqrt{\lambda_{11}} \approx 1013.9$ $\sqrt{\lambda_{12}} \approx 1013.9$

Figure 3.13: 2D Biharmonic eigenvalues B.V.P. with C.P. boundary conditions on a disk with five holes: eigenvalues closest to 1000^2 with their eigenfunctions

3.2.3 Mixed boundary conditions

In this sedtion we mixed the Cahn-Hilliard, Clamped Plate and Simply supported plate boundary conditions in some examples.

Application on $\Omega = [0, L] \times [0, H]$

We choose a C.P. boundary condition on Γ_1 (left boundary), a S.S.P. boundary condition on Γ_2 (right boundary) and a C.H. boundary condition on Γ_3 and Γ_4 (bottom and top boundaries). We represent in Figure 3.14 the twelve first eigenvectors obtained by using the command given in Listing 3.14.

```
pde=fc_vfemp1.PDE(Gop);
bvp=fc_vfemp1.BVP(Th,pde);
for lab=Th.get_labels(1)
  bvp.setDirichlet( lab, 0.,[],1);
  bvp.setRobin( lab, 0.,[],2);
end
[eVec,eVal,flag]=fc_vfemp1.addon.eigs.solve(bvp,'Bop',Bop,'neigs',12,'sigma',0);
```

Listing 3.13: Setting biharmonic eigenvalue problem for plate vibration with simply supported plate boundary conditions $\Omega = [0, 1] \times [0, 1]$. Using operators defined in Listing 3.2.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'CP',1,'SSP',2,'L',1,'H',1,'sigma',0,'neigs',12,'title',false);
```

Listing 3.14: Biharmonic eigenvalues B.V.P. with mixed boundary conditions on the unit square

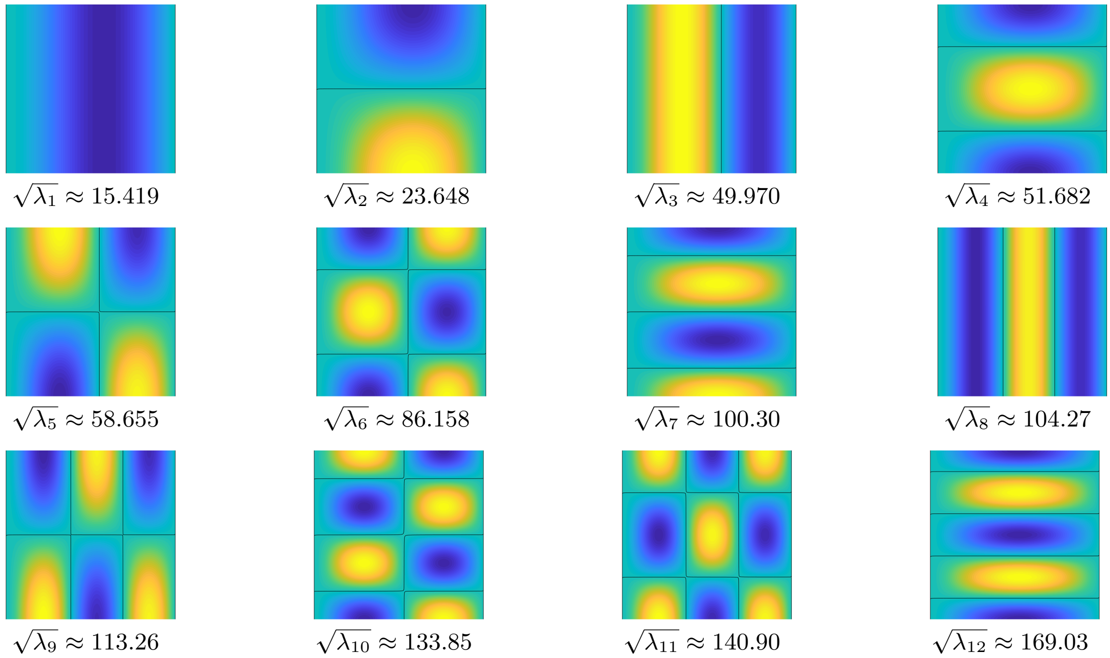


Figure 3.14: 2D Biharmonic eigenvalues B.V.P. with mixed boundary conditions on the unit square: smallest eigenvalues with their eigenfunctions

We represent in Figure 3.15 the twelve eigenvectors associated with the eigenvalues closest to 1000^2 and obtained by using the command given in Listing 3.15.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'CP',1,'SSP',2,'L',1,'H',1,'sigma',0,'neigs',12,'title',false);
```

Listing 3.15: Biharmonic eigenvalues B.V.P. with mixed boundary conditions on the unit square

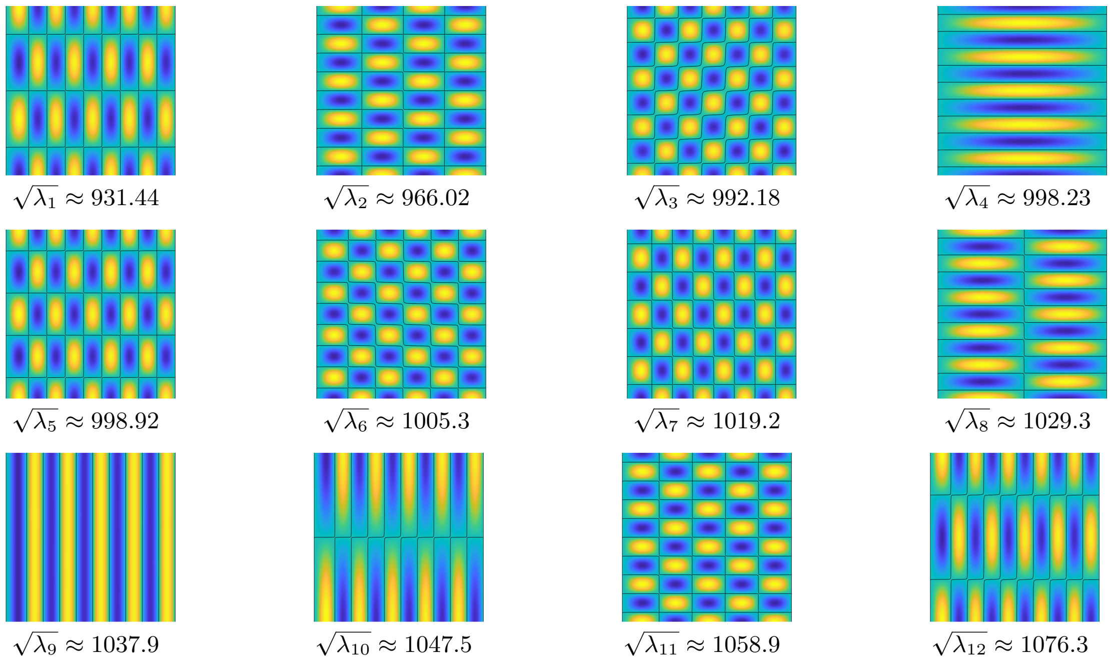


Figure 3.15: 2D Biharmonic eigenvalues B.V.P. with mixed boundary conditions on the unit square: eigenvalues closest to 1000^2 with their eigenfunctions

Application on a disk with 5 holes

We choose a C.P. boundary condition on Γ_{10} (centered hole boundary), a S.S.P. boundary condition on $\Gamma_{20} \cup \Gamma_{21} \cup \Gamma_{22} \cup \Gamma_{23}$ (all the 4 small holes boundary) and a C.H. boundary condition on Γ_1 (big disk exterior boundary). The disk with five holes is given in Figure 2.9. We represent in Figure 3.16 the twelve first eigenvectors obtained by using the command given in Listing 3.16.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'geofile','disk5holes','CP',10,'SSP',[20:23],'sigma',0,'neigs',12, ...
    'title',false);
```

Listing 3.16: Biharmonic eigenvalues B.V.P. with mixed boundary conditions on a disk with five holes

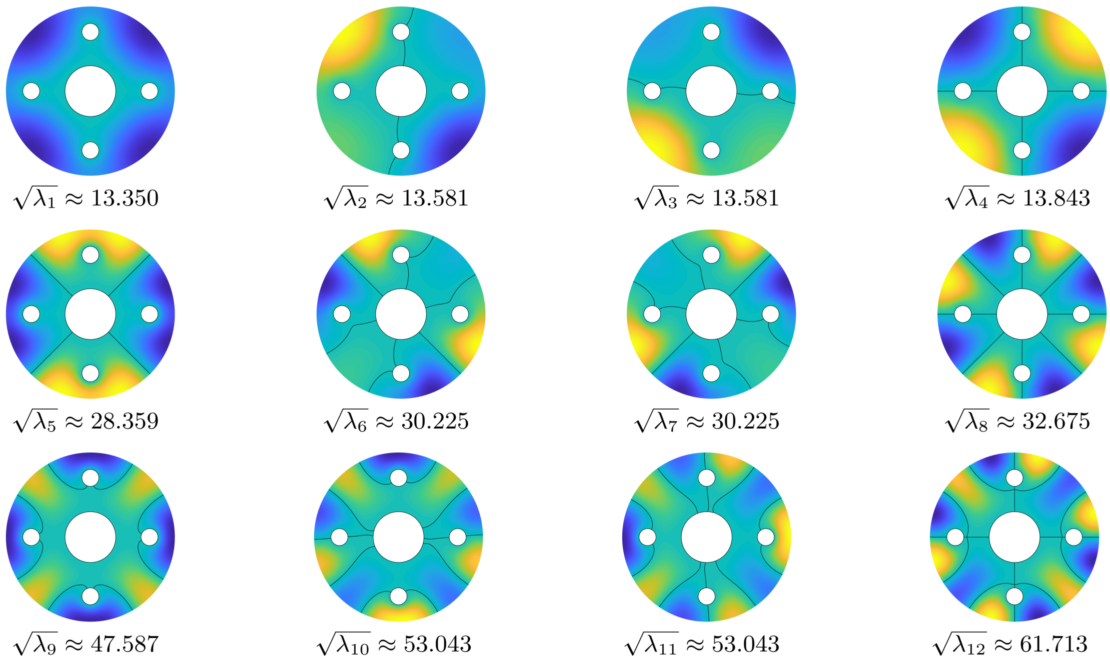


Figure 3.16: 2D Biharmonic eigenvalues B.V.P. with mixed boundary conditions on a disk with five holes: smallest eigenvalues with their eigenfunctions

We represent in Figure 3.17 the twelve eigenvectors associated with the eigenvalues closest to 1000^2 and obtained by using the command given in Listing 3.17.

```
[Th,eVec,lambda,info]=fc_vfemp1.addon.eigs.demos.dim2d.funBiharmonic(200, ...
    'geofile','disk5holes','CP',10,'SSP',[20:23],'sigma',0,'neigs',12, ...
    'title',false);
```

Listing 3.17: Biharmonic eigenvalues B.V.P. with mixed boundary conditions on a disk with five holes

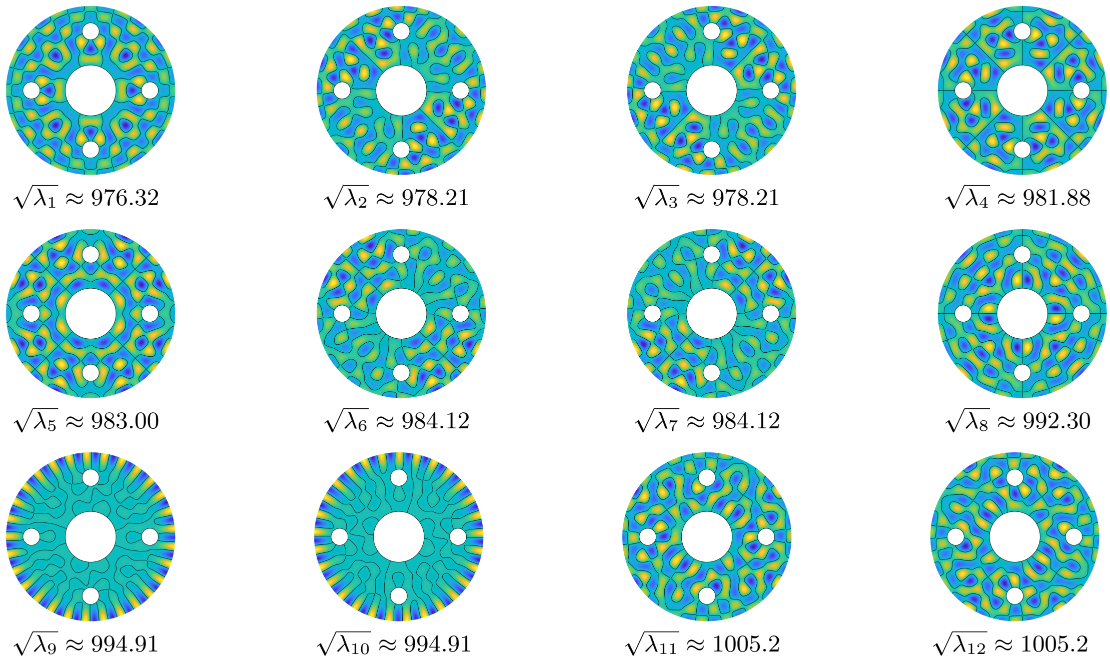


Figure 3.17: 2D Biharmonic eigenvalues B.V.P. with mixed boundary conditions on a disk with five holes: eigenvalues closest to 1000^2 with their eigenfunctions

Chapter A

Biharmonic BVP

Let $\Omega \subset \mathbb{R}^{\dim}$ and $\Gamma = \partial\Omega$. The biharmonic equation is the fourth-order partial PDE given by

$$\Delta^2 u = f, \quad \text{in } \Omega \tag{A.1}$$

where $\Delta^2 u = \Delta(\Delta u) = \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^4 u}{\partial x_i^2 \partial x_j^2} = \sum_{i=1}^{d} \frac{\partial^4 u}{\partial x_i^4} + 2\sum_{i=1}^{d}\sum_{j=i+1}^{d} \frac{\partial^4 u}{\partial x_i^2 \partial x_j^2}$

The boundary conditions on Γ can be

- *Clamped Plate* (CP) or *pure Dirichlet* type:

$$u = \frac{\partial u}{\partial \boldsymbol{n}} = g \tag{A.2}$$

- *Simply Supported Plate* (SSP) or *Navier* type :

$$u = \Delta u = g \tag{A.3}$$

- *Pure Hinged Plate* (PHP) or *Steklov* type :

$$u = \Delta u - (1-\sigma)K\frac{\partial u}{\partial \boldsymbol{n}} = g \tag{A.4}$$

- *Cahn-Hilliard* (CH) type

$$\frac{\partial u}{\partial n} = \frac{\partial \Delta u}{\partial n} = g \tag{A.5}$$

A.0.1 Link with $\mathcal{H}$-operator and boundary conditions

Classically the fourth-order PDE (A.1) is converted to the two second-order PDE

$$-\Delta u = v \tag{A.6}$$
$$-\Delta v = f \tag{A.7}$$

These two equations can be equivalently written as

$$\mathcal{G}\begin{pmatrix}u\\v\end{pmatrix} = \begin{pmatrix}f\\0\end{pmatrix} \quad \text{or} \quad \mathcal{K}\begin{pmatrix}u\\v\end{pmatrix} = \begin{pmatrix}0\\f\end{pmatrix} \tag{A.8}$$

where $\mathcal{G}$ and $\mathcal{K}$ are the $\mathcal{H}$-operators defined by

$$\mathcal{G} = \begin{pmatrix} 0 & \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0} \\ \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0} & \mathcal{L}_{\mathbb{O},\mathbf{0},\mathbf{0},-1}\end{pmatrix} \quad \text{and} \quad \mathcal{K} = \begin{pmatrix} \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0} & \mathcal{L}_{\mathbb{O},\mathbf{0},\mathbf{0},-1} \\ \mathcal{L}_{\mathbb{O},\mathbf{0},\mathbf{0},0} & \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0}\end{pmatrix} \tag{A.9}$$

The Matlab code using the FC-VFEMP1 toolboxto create the operators $\mathcal{G}$ and $\mathcal{K}$ are respectively given in Listings A.1 and A.2.

```
dim=3;d=3;m=2;
A=cell(dim,dim);for i=1:dim,A{i,i}=1;end
Lop=fc_vfemp1.Loperator(dim,d,A,[],[],[]);
Gop=fc_vfemp1.Hoperator(dim,d,m);
Gop.set(1,2,Lop);
Gop.set(2,1,Lop);
Lop2=fc_vfemp1.Loperator(dim,d,[],[],[], 1));
Gop.set(1,1,Lop2);
```

Listing A.1: $\mathcal{G}$ operator with the FC-VFEMP1 toolbox

```
dim=3;d=3;m=3;
A=cell(dim,dim);for i=1:dim,A{i,i}=1;end
Lop=fc_vfemp1.Loperator(dim,d,A,[],[],[]);
Hop=Hoperator(dim,d,m);
Hop.set(1,1,Lop);
Hop.set(2,2,Lop);
Lop2=fc_vfemp1.Loperator(dim,d,[],[],[], 1));
Hop.set(2,2,Lop2);
```

Listing A.2: $\mathcal{K}$ operator with the FC-VFEMP1 toolbox

Let $\boldsymbol{w} = (u, v)$. From (A.8), the components of the conormal derivative of $\boldsymbol{w}$ defined in (1.17) are given by

$$\begin{aligned}\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{K}_1}} &\overset{\text{def}}{=} \sum_{\beta=1}^{2} \frac{\partial \boldsymbol{w}_\beta}{\partial n_{\mathcal{K}_{1,\beta}}} = \sum_{\beta=1}^{2} \left\langle \mathbb{A}^{1,\beta}\,\boldsymbol{\nabla}\,\boldsymbol{w}_\beta, \boldsymbol{n}\right\rangle - \left\langle \boldsymbol{b}^{1,\beta}\boldsymbol{u}_\beta, \boldsymbol{n}\right\rangle \\ &= \langle \mathbb{I}\,\boldsymbol{\nabla}\,\boldsymbol{w}_1, \boldsymbol{n}\rangle = \langle \boldsymbol{\nabla}\,u, \boldsymbol{n}\rangle \\ &= \frac{\partial u}{\partial \boldsymbol{n}} = \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_2}}\end{aligned} \tag{A.10}$$

and

$$\begin{aligned}\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{K}_2}} &\overset{\text{def}}{=} \sum_{\beta=1}^{2} \frac{\partial \boldsymbol{w}_\beta}{\partial n_{\mathcal{K}_{2,\beta}}} = \sum_{\beta=1}^{2} \left\langle \mathbb{A}^{2,\beta}\,\boldsymbol{\nabla}\,\boldsymbol{w}_\beta, \boldsymbol{n}\right\rangle - \left\langle \boldsymbol{b}^{2,\beta}\boldsymbol{u}_\beta, \boldsymbol{n}\right\rangle \\ &= \langle \mathbb{I}\,\boldsymbol{\nabla}\,\boldsymbol{w}_2, \boldsymbol{n}\rangle = \langle \boldsymbol{\nabla}\,v, \boldsymbol{n}\rangle \\ &= \frac{\partial v}{\partial \boldsymbol{n}} = \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_1}}\end{aligned} \tag{A.11}$$

From *Vector* BVP (1.14)-(1.16), using $\mathcal{G}$ operator one can impose the following boundary conditions with $\Gamma_\alpha^D \cap \Gamma_\alpha^R = \emptyset$, $\forall \alpha \in [\![1,2]\!]$,

$$\begin{aligned}\boldsymbol{w}_\alpha &= g_\alpha^D && \text{on } \Gamma_\alpha^D,\ \forall \alpha \in [\![1,2]\!],\\ \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_\alpha}} + a_\alpha^R \boldsymbol{w}_\alpha &= g_\alpha^R && \text{on } \Gamma_\alpha^R,\ \forall \alpha \in [\![1,2]\!]\end{aligned}$$

i.e.

$$\begin{aligned} u &= g_1^D && \text{on } \Gamma_1^D, & v &= g_2^D && \text{on } \Gamma_2^D,\\ \frac{\partial v}{\partial \boldsymbol{n}} + a_1^R u &= g_1^R && \text{on } \Gamma_1^R, & \frac{\partial u}{\partial \boldsymbol{n}} + a_2^R v &= g_2^R && \text{on } \Gamma_2^R\end{aligned}$$

With $\mathcal{K}$ operator one can impose the following boundary conditions

$$\begin{aligned}\boldsymbol{w}_\alpha &= g_\alpha^D && \text{on } \Gamma_\alpha^D,\ \forall \alpha \in [\![1,2]\!],\\ \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{K}_\alpha}} + a_\alpha^R \boldsymbol{w}_\alpha &= g_\alpha^R && \text{on } \Gamma_\alpha^R,\ \forall \alpha \in [\![1,2]\!]\end{aligned}$$

i.e.

$$\begin{aligned} u &= g_1^D && \text{on } \Gamma_1^D, & v &= g_2^D && \text{on } \Gamma_2^D,\\ \frac{\partial u}{\partial \boldsymbol{n}} + a_1^R u &= g_1^R && \text{on } \Gamma_1^R, & \frac{\partial v}{\partial \boldsymbol{n}} + a_2^R v &= g_2^R && \text{on } \Gamma_2^R\end{aligned}$$

remark A.1

One can neither impose *clamped plate* (A.2) nor *Pure Hinged Plate* (A.4) boundary conditions with $\mathcal{K}$ operator. This is why thereafter we will only use the $\mathcal{G}$ operator.

In the same way, with the operator $\mathcal{G}$ given in (A.8), the components of the conormal derivative of $\boldsymbol{w}$ defined in (1.17) are given by

$$\begin{aligned}\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_1}} &\stackrel{\text{def}}{=} \sum_{\beta=1}^{2}\frac{\partial \boldsymbol{w}_\beta}{\partial n_{\mathcal{G}_{1,\beta}}} = \sum_{\beta=1}^{2}\left\langle \mathbb{A}^{1,\beta}\,\boldsymbol{\nabla}\,\boldsymbol{w}_\beta, \boldsymbol{n}\right\rangle - \left\langle \boldsymbol{b}^{1,\beta}\boldsymbol{u}_\beta, \boldsymbol{n}\right\rangle \\ &= \left\langle \mathbb{I}\,\boldsymbol{\nabla}\,\boldsymbol{w}_2, \boldsymbol{n}\right\rangle = \left\langle \boldsymbol{\nabla}\,v, \boldsymbol{n}\right\rangle \\ &= \frac{\partial v}{\partial \boldsymbol{n}}\end{aligned} \tag{A.12}$$

and

$$\begin{aligned}\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_2}} &\stackrel{\text{def}}{=} \sum_{\beta=1}^{2}\frac{\partial \boldsymbol{w}_\beta}{\partial n_{\mathcal{G}_{2,\beta}}} = \sum_{\beta=1}^{2}\left\langle \mathbb{A}^{2,\beta}\,\boldsymbol{\nabla}\,\boldsymbol{w}_\beta, \boldsymbol{n}\right\rangle - \left\langle \boldsymbol{b}^{2,\beta}\boldsymbol{u}_\beta, \boldsymbol{n}\right\rangle \\ &= \left\langle \mathbb{I}\,\boldsymbol{\nabla}\,\boldsymbol{w}_1, \boldsymbol{n}\right\rangle = \left\langle \boldsymbol{\nabla}\,u, \boldsymbol{n}\right\rangle \\ &= \frac{\partial u}{\partial \boldsymbol{n}}\end{aligned} \tag{A.13}$$

Let us denotes the two partitions avec the boundary Γ:

$$\mathring{\Gamma}_\alpha^D \cap \mathring{\Gamma}_\alpha^R = \emptyset \quad \text{and} \quad \Gamma_\alpha^D \cup \Gamma_\alpha^R = \Gamma, \quad \forall \alpha \in [\![1,2]\!] \tag{A.14}$$

From *Vector* BVP (1.14)-(1.16), using $\mathcal{G}$ operator one can impose the following boundary conditions

$$\begin{aligned}\boldsymbol{w}_\alpha &= g_\alpha^D && \text{on } \Gamma_\alpha^D,\ \forall \alpha \in [\![1,2]\!],\\ \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_\alpha}} + a_\alpha^R \boldsymbol{w}_\alpha &= g_\alpha^R && \text{on } \Gamma_\alpha^R,\ \forall \alpha \in [\![1,2]\!]\end{aligned}$$

i.e.

$$\begin{aligned}u &= g_1^D && \text{on } \Gamma_1^D, & v &= g_2^D && \text{on } \Gamma_2^D,\\ \frac{\partial v}{\partial \boldsymbol{n}} + a_1^R u &= g_1^R && \text{on } \Gamma_1^R, & \frac{\partial u}{\partial \boldsymbol{n}} + a_2^R v &= g_2^R && \text{on } \Gamma_2^R\end{aligned}$$

To resume, we give a generic biharmonic BVP using the operator $\mathcal{G}$ in *Vector* BVP (3) which is equivalent to the generic mixed formulation for the biharmonic BVP given in *Vector* BVP (4)

Vector BVP 3 : generic biharmonic BVP with $\mathcal{G}$ operator

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

$$\mathcal{G}(\boldsymbol{w}) = \begin{pmatrix} f \\ 0 \end{pmatrix} \quad \text{in } \Omega, \tag{A.15}$$

$$\boldsymbol{w}_1 = g_1^D \quad \text{on } \Gamma_1^D, \qquad \boldsymbol{w}_2 = g_2^D \quad \text{on } \Gamma_2^D, \tag{A.16}$$

$$\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_1}} + a_1^R \boldsymbol{w}_1 = g_1^R \quad \text{on } \Gamma_1^R, \qquad \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_2}} + a_2^R \boldsymbol{w}_2 = g_2^R \quad \text{on } \Gamma_2^R. \tag{A.17}$$

Vector BVP 4 : generic mixed formulation for the biharmonic BVP

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

$$\begin{pmatrix} 0 & \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0} \\ \mathcal{L}_{\mathbb{I},\mathbf{0},\mathbf{0},0} & \mathcal{L}_{\mathbb{O},\mathbf{0},\mathbf{0},-1} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \quad \text{in } \Omega, \tag{A.18}$$

$$u = g_1^D \quad \text{on } \Gamma_1^D, \qquad v = g_2^D \quad \text{on } \Gamma_2^D, \tag{A.19}$$

$$\frac{\partial v}{\partial \boldsymbol{n}} + a_1^R u = g_1^R \quad \text{on } \Gamma_1^R, \qquad \frac{\partial u}{\partial \boldsymbol{n}} + a_2^R v = g_2^R \quad \text{on } \Gamma_2^R. \tag{A.20}$$

It's very easy to write (A.15) (or (A.18)) from the generic formulation of the biharmonic BVP with $\mathcal{G}$ operator with the FC-VFEMP1 toolbox: the source code is given in Listing A.3 where Th is a given fc_vfemp1.siMesh object and f a given Matlab function or scalar.

```
dim=2;d=2;m=2;
Lop=fc_vfemp1.Loperator(dim,d,{1,0;0,1},[],[],[]);
Gop=fc_vfemp1.Hoperator(dim,d,m);
Gop.set(1,2,Lop);
Gop.set(2,1,Lop);
Lop2=fc_vfemp1.Loperator(dim,d,[],[],[], 1));
Gop.set(1,1,Lop2);
pde=fc_vfemp1.PDE(Gop,{f,0})
bvp=fc_vfemp1BVP(Th,pde)
```

Listing A.3: Writing the $\mathcal{G}(\boldsymbol{w}) = \begin{pmatrix} f \\ 0 \end{pmatrix}$ with the FC-VFEMP1 package

remark A.2

By default the boundary conditions of a fc_vfemp1.BVP object are set to **homogeneous Neumann** so we have

$$\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_1}} = \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_2}} = 0 \quad \Leftrightarrow \quad \frac{\partial v}{\partial \boldsymbol{n}} = \frac{\partial u}{\partial \boldsymbol{n}} = 0$$

which is the homogeneous **Cahn-Hilliard** boundary condition.

Now we will see in the next sections how to set the **Clamped Plate** (CP), **Simply Supported Plate** (CP) and **Cahn-Hilliard** (CH) boundary conditions.

A.1 Some boundary conditions

Let $\Gamma_{\text{lab}} \subset \Gamma$. We want to set one of the following boundary conditions on Γ_{lab}:

- *Clamped Plate* (CP) or *pure Dirichlet* type:

$$u = g_a \ \ \textbf{and} \ \ \frac{\partial u}{\partial \boldsymbol{n}} = g_b \quad \text{on } \Gamma_{\text{lab}} \tag{A.21}$$

- *Simply Supported Plate* (SSP) or *Navier* type :

$$u = g_a \ \ \textbf{and} \ \ \Delta u = -g_b \quad \text{on } \Gamma_{\text{lab}} \tag{A.22}$$

- *Cahn-Hilliard* (CH) type

$$\frac{\partial u}{\partial n} = g_a \ \ \textbf{and} \ \ \frac{\partial \Delta u}{\partial n} = -g_b \quad \text{on } \Gamma_{\text{lab}} \tag{A.23}$$

Now we will see how to rewrite theses boundary conditions as those in *Vector* BVP 4, equations (A.19)-(A.20), and *Vector* BVP 3, equations (A.16)-(A.17).

A.1.1 Clamped Plate boundary condition

From (A.19) and (A.20), we deduce that (A.21) imposes

$$u = g_a \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_1^D \quad \text{and} \quad \frac{\partial u}{\partial \boldsymbol{n}} = g_b \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_2^R$$

So with $\mathcal{G}$ operator and with $\boldsymbol{w} = (u, v)$ we obtain

$$w_1 = g_a \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_1^D \quad \text{and} \quad \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_2}} = g_b \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_2^R$$

Let bvp be the fc_vfemp1.BVP object build in Listing A.3. We want to set this object with the (CP) boundary condition. To set the dirichlet condition on first component $w_1 = g_a$ on Γ_{lab}, we can use the setDirichlet method of the fc_vfemp1.BVP object:

```
bvp.setDirichlet(lab,ga,1)
```

To set the Neumann condition on second component $\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_2}} = g_b$ we can use the setRobin method of the fc_vfemp1.BVP object:

```
bvp.setRobin(lab,gb,2)
```

A.1.2 Simply Supported Plate boundary condition

From (A.19) and (A.20), we deduce that (A.22) imposes

$$u = g_a \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_1^D \quad \text{and} \quad v = g_b \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_2^D$$

So with $\mathcal{G}$ operator and with $\boldsymbol{w} = (u, v)$ we obtain

$$w_1 = g_a \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_1^D \quad \text{and} \quad w_2 = g_b \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_2^D$$

Let bvp be the fc_vfemp1.BVP object build in Listing A.3. We want to set this object with the (SSP) boundary condition. To set the dirichlet conditions, we can use the setDirichlet method of the fc_vfemp1.BVP object:

```
bvp.setDirichlet(lab,ga,1)
bvp.setDirichlet(lab,gb,2)
```

or

```
bvp.setDirichlet(lab,[ga,gb])
```

A.1.3 Cahn-Hilliard boundary condition

From (A.19) and (A.20), we deduce that (A.23) imposes

$$\frac{\partial v}{\partial n} = g_b \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_1^R \quad \text{and} \quad \frac{\partial u}{\partial \boldsymbol{n}} = g_a \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_2^R$$

where $v = -\Delta u$.

So with $\mathcal{G}$ operator and with $\boldsymbol{w} = (u, v)$ we obtain

$$\frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_1}} = g_b \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_1^R \quad \text{and} \quad \frac{\partial \boldsymbol{w}}{\partial n_{\mathcal{G}_2}} = g_a \quad \text{on } \Gamma_{\text{lab}} \subset \Gamma_2^R$$

Let bvp be the fc_vfemp1.BVP object build in Listing A.3. We want to set this object with the (CH) boundary condition. To set the Neumann conditions, we can use the setRobin method of the fc_vfemp1.BVP object:

```
bvp.setRobin(lab,gb,[],1)
bvp.setRobin(lab,ga,[],2)
```

or more concisely

```
bvp.setRobin(lab,{gb,ga})
```

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

[3] L. Fox, P. Henrici, and C. Moler. Approximations and bounds for eigenvalues of elliptic operators. *SIAM Journal on Numerical Analysis*, 4(1):89–102, 1967.

[4] J. M. Gedicke. *On the Numerical Analysis of Eigenvalue Problems*. PhD thesis, University of Berlin, 2013.

[5] D. S. Grebenkov and B.-T. Nguyen. Geometrical structure of laplacian eigenfunctions. *SIAM Review*, 55(4):601–667, jan 2013.

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

[7] L. N. Trefethen and T. Betcke. Computed eigenmodes of planar regions. *Manchester Institute for Mathematical Sciences School of Mathematics*, 2006.

[8] Quan Yuan and Zhiqing He. Bounds to eigenvalues of the laplacian on l-shaped domain by variational methods. *Journal of Computational and Applied Mathematics*, 233(4):1083 – 1090, 2009.

Informations for git maintainers of the vfem$\mathbb{P}_1$-eigs Matlab toolbox

git informations on the toolboxes used to build this manual

```
--------------------------------------------------
 Addon eigs of the fc-vfemp1 package
--------------------------------------------------
  name : eigs
   tag : 0.1.2
commit : f5c2a35cf4268ca3727ba459f814305344113d0b
  date : 2020-04-05
  time : 10-10-14
status : 0
--------------------------------------------------
  name : fc-vfemp1
   tag : 0.2.2
commit : 881e3df2cc277b3bd9db00e91b9fd45fc4e341d3
  date : 2020-04-04
  time : 13-27-22
status : 0
--------------------------------------------------
--------------------------------------------------
 Packages used by the fc-vfemp1 package
--------------------------------------------------
  name : fc-tools
   tag : 0.0.32
commit : 7539b1a386933d564cfc07ee5428605418c4ba7e
  date : 2020-04-01
  time : 16-02-26
status : 0
--------------------------------------------------
  name : fc-bench
   tag : 0.1.2
commit : 666dc60d1277f5fa9c99dee4ae1c33270f22c57d
  date : 2020-02-16
  time : 06-38-46
status : 0
--------------------------------------------------
  name : fc-hypermesh
   tag : 1.0.3
commit : c520b34cfd7eb0dbf9e4ecd459fd7162db73cc58
  date : 2020-02-16
  time : 08-34-19
status : 0
--------------------------------------------------
  name : fc-amat
   tag : 0.1.2
commit : 957340f6e71d805dbd8b9d04c434b24fd3f92591
  date : 2020-02-16
  time : 06-39-42
status : 0
--------------------------------------------------
  name : fc-meshtools
   tag : 0.1.3
commit : cdbc41bc98af4e4faccc1746024aced1f21aae53
  date : 2020-02-17
  time : 10-52-56
status : 0
--------------------------------------------------
  name : fc-graphics4mesh
   tag : 0.1.4
commit : 0bc57da35958c469ac43b56f55968233d720206c
  date : 2020-04-01
  time : 07-22-06
status : 0
--------------------------------------------------
  name : fc-oogmsh
   tag : 0.2.3
commit : 5da0f93c4701b185dda7da047d015ddaa8119026
  date : 2020-03-20
  time : 08-15-41
status : 0
--------------------------------------------------
  name : fc-siplt
   tag : 0.2.3
commit : 3c9ef4a46d80d7898360b18e1db7e795fbae44d3
  date : 2020-04-03
  time : 15-44-34
status : 0
--------------------------------------------------
  name : fc-simesh
   tag : 0.4.3
commit : 108c6e14e74fa6eaae8df726b5723e834945f1b3
  date : 2020-04-04
  time : 14-10-57
status : 0
--------------------------------------------------
```

git informations on the LaTeX package used to build this manual

```
-------------------------------------------------
  name : fctools
   tag :
 commit : c6841f9838937ffe0b2a21ec7162efcc615774e3
  date : 2020-04-03
  time : 10:06:02
 status : 1
-------------------------------------------------
```

Using the remote configuration repository:

url	`ssh://lagagit/MCS/Cuvelier/Matlab/fc-config`
commit	`8e7bd91a47e41e84824fb78be23d01004931a1fc`