UH APPLIED MATHEMATICS MASTER TUTORIAL PROJECT

Steklov Eigenfunctions

Applications to Div-Curl Systems using FreeFEM++

Simpkins, Douglas R

5/19/2014

Prepared for
Dr. Auchmuty

Contents

1. Introduction

This project investigates the application of the Steklov-eigenvalue expansion method (SEM) to two types of boundary value problems. The first type of problem is a mixed Dirichlet-Neumann boundary value problem (mixed DN bvp) involving a second-order uniformly elliptic equation subjected to inhomogeneous Dirichlet data on part of the boundary and homogenous Neumann flux data on the remainder of the boundary. The second type of problem is a div-curl system with prescribed flux data on the boundary (N div-curl sys). Both problem types model various equilibrium phenomena in physics with applications ranging from electrostatics, magnetostatics, Newtonian gravity, heat conduction, diffusion, and ideal inviscid fluid flow. Mixed DN bvps are scalar equations that model phenomenon where values of the underlying potential function are known on the boundary such as temperature in heat conduction. Div-curl systems are vector equations that model systems where either normal flux or tangential flow is known on the boundary. Only normal flux boundary conditions are considered here.

A solution of the mixed DN bvp can be represented by a Steklov-eigenfunction expansion while a solution of the N div-curl sys can be represented by the gradient of a Steklov-eigenfunction expansion. These solution representations are demonstrated and compared against solutions from traditional Galerkin finite element methods (FEM) using the software package FreeFEM++ [1]. In addition, numerical calculation of Steklov-eigenvalues and eigenfunctions on a rectangle are compared against analytical formula derived using the method of separation of variables. Some of the advantages and disadvantages of SEM are also discussed. Additionally, some examples of Steklov-spectra and associated eigenfunctions for specific geometries are shown.

In order to minimally demonstrate SEM, relevant pieces of the mathematical framework, existence results and variational forms are taken verbatim from Auchmuty [2], [3], and [4]. The problem formulation and well-posedness of the div-curl system, the mixed Dirichlet-Neumann boundary value problem and Steklov-eigenfunction expansion method are described in detail in [2], [3] and [4], respectively. Representing solutions of the A-harmonic div-curl system as the gradient of a Steklov-eigenfunction expansion is a novel idea due to Dr. Auchmuty and no known literature exists describing its soundness. No formal proof of the result is given in this work but instead, results of its application are demonstrated.

2. Mixed Dirichlet Neumann Boundary Value Problem (mixed DN bvp)

The mixed DN bvp is a self-adjoint and second order equation of the form:

$$div\big(A(x)\nabla u(x)\big) = 0, \;\; x \in \Omega \tag{1}$$

subject to the mixed Dirichlet and Neumann boundary conditions

$$u(x) = g(x), \;\; x \in \Sigma \quad \text{and} \quad A(x)\nabla u(x) = 0, \;\; x \in \partial\Omega \backslash \Sigma \tag{2}$$

where the boundary interface $\Sigma \cap \partial\Omega \backslash \Sigma$ has length or area zero and $\Sigma, \partial\Omega$ have finite measure. Problem (1)-(2) is referred to as a mixed DN A-Harmonic equation and models steady-state heat conduction and diffusion type problems absent of sources and sinks. When the Dirichlet boundary

condition in (2) is removed, problem (1)-(2) is referred to as a Neumann A-Harmonic equation (NA-Harmonic). The NA-Harmonic equation plays a role in calculating solutions to the div-curl sys. The solution to the NA-Harmonic equation is not unique; two solutions differ by a constant. If the set of solutions are restricted to solutions with zero mean, i.e. $\frac{1}{|\Omega|}\int_\Omega \varphi \ dx = 0$, then a unique solution to the NA-Harmonic problem can be sought, see 6.2.3-6.2.4 in [5].

3. Div-Curl System with Prescribed Flux on the Boundary (N div-curl sys)

A div-curl system has the form

$$div\big(A(x)v(x)\big) = \rho(x), \tag{3}$$
$$curl\big(v(x)\big) = \omega(x), \quad x \in \Omega \tag{4}$$

subject to the prescribed flux on the boundary

$$(A(x)\mathrm{v}(x)) \cdot \upsilon = \mu(x), \ \ x \in \partial\Omega. \tag{5}$$

When ρ and ω are zero, and $A(x)$ is the identity matrix, equations (3)-(4) are referred to as a Laplacian vector field (or Harmonic vector field). When $A(x)$ is a symmetric positive-definite (spd) matrix, solutions v of (3)-(4) are said to be both A-solenoidal and irrotational, or equivalently A-harmonic. When ρ is a real constant and ω is a real (vector) constant and $A(x)$ is spd and independent of x, then (3), (4), and (5) can be converted to an equivalent A-Harmonic vector field. To see this for two and three dimensional vector fields, let $M_s = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}$ and $M_s = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$, respectively assuming $\omega = (\omega_1, \omega_2, \omega_3)$ is a vector in the three dimensional case. Notice, M_s is an anti-symmetric matrix, i.e. $M_s = -M_s^T$. Furthermore, define $M = \frac{\rho}{3}A^{-1} + M_s$, which is well-defined as A^{-1} exists as A is spd. For symmetric A and anti-symmetric M_s, $div(Ax) = tr(A)$ and $tr(AM_s) = 0$. Let $\hat{v}(x) = v(x) - Mx$. It follows that

$$div\big(Av(x)\big) = \rho \rightarrow div\big(A\hat{v}(x)\big) = 0 \ \ , \quad x \in \Omega. \tag{6}$$

Similarly, for symmetric A, $curl\, Ax = 0$ hence

$$curl\big(v(x)\big) = \omega \rightarrow curl\big(\hat{v}(x)\big) = 0 \ \ , \quad x \in \Omega. \tag{7}$$

The prescribed flux boundary condition (5) can be modified to

$$(A\hat{v}(x)) \cdot \upsilon = \mu(x) - (AMx) \cdot \upsilon, \ \ x \in \partial\Omega. \tag{8}$$

The div-curl system in (6)-(8) models a diverse number of steady phenomena in physics. The table below summarizes several such systems with constant circulation and/or constant divergence.

Steady Phenomenon	**Primary Field** $\boldsymbol{\delta}$	**Secondary Field** $\boldsymbol{\gamma}$	**Constitutive Matrix** ε	**Constant Source/Sink** ρ	**Constant Circulation** $\boldsymbol{\omega}$
Heat Conduction	Heat flux $\boldsymbol{q}$	Thermal flux $\boldsymbol{s}$	Thermal Conductivity Matrix $-k$	Heat Generation q_s	None
Diffusive Transport	Material flux $\boldsymbol{j}$	Concentration flux $\boldsymbol{c}$	Diffusivity Matrix $-d$	None	None
Electrostatics	Displacement field $\boldsymbol{D}$	Electric Field $\boldsymbol{E}$	Permittivity Matrix ε	Free-Charge Density q_f	None
Magnetostatics	Magnetic field $\boldsymbol{B}$	Magnetic Intensity Field $\boldsymbol{H}$	Magnetic Permeability Matrix μ	None	Current Density $\boldsymbol{J}$
Ideal Inviscid Fluid Flow	Fluid Momentum $\boldsymbol{m}$	Fluid Velocity $\boldsymbol{v}$	Density Matrix ρ	None	Vorticity (rigid rotation) $\boldsymbol{\omega}$
Conservative Newtonian Gravity	Gravitational Displacement $\boldsymbol{g_D}$	Gravitational Field $\boldsymbol{g}$	Gravitational Matrix G^{-1}	Mass Density of Object ρ	None

Table 1 – various applications of the A-Harmonic vector field.

4. **Mathematical Framework, Assumptions and Variational Forms**

Under certain assumptions on the interior and boundary of the geometry, on the functions $A(x)$, $\rho(x)$ and $\omega(x)$, and on the boundary conditions, the mixed DN bvp (1)-(2) and N div-curl sys (3)-(5) are shown to have unique finite energy solutions in Auchmuty [2] and [3], respectively. This section captures relevant mathematical facts that are used to frame problems (1)-(2) and (3)-(5) into equivalent variational problems that are conducive for computation using finite element approximation techniques.

(i) Assumptions on Geometry

A *region* is a non-empty, connected, open subset of $\mathbb{R}^n$. The closure of Ω is denoted as $\overline{\Omega}$ and its boundary by $\partial\Omega \coloneqq \overline{\Omega}\backslash\Omega$. In order for solutions to exist for problems (1)-(2) and (3)-(5), the following condition on geometry is required:

(B1) Ω is a bounded region contained in $\mathbb{R}^n$, for $n \geq 2$ (for div-curl systems,$n = 2$ is only considered) whose boundary $\partial\Omega$ is the union of a finite number of disjoint closed Lipschitz surfaces (curves); each having finite surface area (arc length).

This regularity condition is necessary for the existence of a unit outward normal to be well-defined almost everywhere on the boundary $\partial\Omega$. Consequentially, the trace map Γ is a well-defined compact operator that is important for the application of Steklov-eigenfunction expansions. The region is also assumed to either be a simple region or a region that contains J holes, where each hole is a smooth curve. These holes influence the solution of the div-curl systems problem.

(B2) The set Ω satisfies (B1) and $\partial\Omega$ consists of $J + 1$ disjoint, simple closed Lipschitz curves γ_j, each of finite length.

In addition, for the mixed DN bvp, the boundary is the union of two subsets, $\partial\Omega = \Sigma \cup \overline{\Sigma}$. The following condition is imposed on this boundary:

(B3) Σ is a non-empty open subset of $\partial\Omega$, Σ and $\overline{\Sigma}$ both have positive surface area (arc length) and the intersection (interface) $\Sigma \cap \overline{\Sigma}$ is a set of measure zero.

(ii) Definitions , Notation and Assumptions on Matrix A

Derivatives of $u(x)$ are taken in the weak sense and are denoted by $\frac{\partial u}{\partial x}$. When $u, v \in \mathbb{R}^2$, the components are given by $\mathrm{u} = (u_1, u_2)$ and $\mathrm{v} = (v_1, v_2)$. The usual Euclidean inner product and norm are given by $u \cdot v$ and $|u|$. When $u: \Omega \to \mathbb{R}^2$ is a vector field, the *divergence* and *curl* are the scalar-valued functions on Ω defined by

$$div\, u \coloneqq u_{1,1}(x) - u_{2,2}(x)\,,$$
$$curl\, u \coloneqq u_{2,1}(x) - u_{1,2}(x)\,,$$

respectively. Here the u_i are the components of u, and $u_{i,j} = \frac{\partial u_i}{\partial x_j}$ is the j-th weak derivative of the u_i. The *gradient* of the function u, $\nabla u: \Omega \to \mathbb{R}^2$, is the vector field given by $\nabla u(x) = (u_{,1}(x), u_{,2}(x))$. The *Curl* of the function u is the vector field defined by $(\nabla \wedge u)(x) = (u_{,2}(x), -u_{,1}(x))$. The matrix $A(x)$ is a symmetric positive definite (spd) $n \times n$ matrix, for each $x \in \overline{\Omega}$. In addition, the matrix must satisfy:

(A1) Each component $a_{i,j}$ of A is a bounded Lebesgue measureable function and there exist positive constants c_0, c_1 such that, for all $x \in \Omega$ and $u \in \mathbb{R}^n$,

$$\boldsymbol{c_0|u|^2 \leq (A(x)u) \cdot u \leq c_1|u|^2.} \qquad (9)$$

(iii) Framework of Function Spaces, Tools and Decompositions Theorems

The space $L_2(\Omega;\mathbb{R}^m)$ with $m = 1$ or 2 is the usual space of Lebesgue square measureable functions. It is a real Hilbert space with the usual norm and inner product denoted by $\|.\|$ and $\langle .,. \rangle$, respectively. The weighted inner product, provided A satisfies (A1), is well-defined and given by

$$\langle u, v\rangle_A := \int_\Omega \big(A(x)u(x)\big)\cdot v(x)\, dx, \tag{10}$$

with associated norm $||.||_A$. $H^1(\Omega) := \{u \in L_2(\Omega) \mid \frac{\partial u_i}{\partial x_j} \in L_2(\Omega)\}$ is the usual standard Sobolev space. $H^1(\Omega)$ is a real Hilbert space under the standard H^1-inner product

$$\langle u, v\rangle_{1,2} := \int_\Omega u(x)v(x) + \nabla u(x)\cdot\nabla v(x)\, dx \tag{11}$$

and the associated norm is given by $\|.\|_{1,2}$. Similarly, $H^1(\Omega;\mathbb{R}^2)$ is a Sobolev space with the usual vector function inner product and associated norm. Notation for the vector inner product and norm is the same notation used for the scalar inner product and norm in (11). The vector field $u \in H^1(\Omega;\mathbb{R}^2)$ is said to be weakly *E-solenoidal* and *weakly irrotational* when

$$\int_\Omega (\mathrm{Au})\cdot\nabla\varphi\, dx = 0 \text{ , for all } \varphi \in C_c^\infty(\Omega), \tag{12}$$

$$\int_\Omega \mathrm{u}\cdot \mathrm{Curl}\ \varphi\, dx = 0, \text{ for all } \varphi \in C_c^\infty(\Omega), \tag{13}$$

respectively, where $C_c^\infty(\Omega)$ is the set of all continuous functions on Ω with compact support. The field u is *weakly A-harmonic* on Ω if it is both weakly E-solenoidal and weakly irrotational. Similarly for scalar valued functions $u \in H^1(\Omega)$, u is said to be *weakly A-Harmonic* provided that

$$\int_\Omega (\mathrm{A}\nabla \mathrm{u})\cdot\nabla\varphi\, dx = 0\ , \quad \text{for all } \varphi \in C_c^\infty(\Omega). \tag{14}$$

Let $H^1_{\Sigma 0}(\Omega) := \{\mathrm{u} \in \mathrm{H}^1(\Omega) | \Gamma u = 0, on\ \Sigma\}$ be the closed subspace of $\mathrm{H}^1(\Omega)$ of all functions with vanishing trace on Σ. It is shown in [3] that the bilinear form $[.,.]_{A\Sigma}: H^1(\Omega)^2 \to \mathbb{R}$ defined by

$$[u, v]_{A\Sigma} = \int_\Omega (\mathrm{A}\nabla \mathrm{u})\cdot\nabla \mathrm{v}\, dx + \int_\Sigma \mathrm{u}\,\mathrm{v}\, d\sigma, \text{ for } u, v \in H^1(\Omega) \tag{15}$$

is an inner product on $\mathrm{H}^1(\Omega)$ that is equivalent to the standard H^1- inner product defined in (10). Let $\langle u, v\rangle_A = \int_\Omega (\mathrm{A}\nabla \mathrm{u})\cdot\nabla \mathrm{v}\, dx$ and $\langle u, v\rangle_\Sigma = \int_\Sigma \mathrm{uv}\, d\sigma$ denote the components whose sum is $[u, v]_{A\Sigma}$. Let $\mathcal{L} = div\big(A(x)\nabla u(x)\big)$ then $[.,.]_{A\Sigma}$ orthogonally decomposes $H^1(\Omega)$ into two subspaces

$$H^1(\Omega) = H^1_{\Sigma 0}(\Omega) \oplus_A ker\mathcal{L}(\Sigma). \tag{16}$$

When $\Sigma = \partial\Omega$, then $ker\mathcal{L}(\Sigma) = \mathcal{H}_A(\Omega)$, the subspace of weakly A-Harmonic functions in $H^1(\Omega)$. This decomposition is fundamental for the SEM method.

Similarly, $L_2(\Omega;\mathbb{R}^2)$ is shown to be decomposed into the following spaces [2]),

$$G(\Omega) \coloneqq \{\nabla\varphi | \varphi \in H^1(\Omega)\},$$

$$C_{A0}(\Omega) \coloneqq \{\mathrm{A}^{-1}\mathrm{Curl}\psi | \psi \in H_0^1(\Omega)\},$$

$\mathcal{H}_{Ev0}(\Omega)$ is the A-orthogonal complement of $G(\Omega) \oplus C_{A0}(\Omega)$.

Let $u \in L_2(\Omega;\mathbb{R}^2)$, then

u is A-orthogonal to $G(\Omega)$ provided that u is A-solenoidal and $\big(A(x)\mathrm{v}(x)\big)\cdot \upsilon = 0$ on $\partial\Omega$,

u is A-orthogonal to $C_{A0}(\Omega)$ provided that u is irrotational,

u is A-orthogonal to $\mathcal{H}_{Ev0}(\Omega)$ provided that u is E-harmonic and $\big(A(x)\mathrm{v}(x)\big)\cdot \upsilon = 0$ on $\partial\Omega$.

As shown in Theorem 3 of [2],

$$L_2(\Omega;\mathbb{R}^2) = G(\Omega) \oplus C_{A0}(\Omega) \oplus \mathcal{H}_{Ev0}(\Omega). \tag{17}$$

Hence, for $u \in L_2(\Omega;\mathbb{R}^2)$, there is a *potential function* $\varphi \in H^1(\Omega)$, *stream function* $\psi \in H_0^1(\Omega)$ and *harmonic vector function* $\mathrm{h} \in \mathcal{H}_{Ev0}(\Omega)$ such that

$$u(x) = \nabla\varphi(x) + A(x)^{-1}Curl\ \psi(\mathrm{x}) + \mathrm{h}(\mathrm{x})\,. \tag{18}$$

The potential function φ uniquely satisfies

$$\int_\Omega (\mathrm{A}\nabla\varphi)\cdot\nabla\chi\ - \rho\chi\,dx = \int_{\partial\Omega}\mu\chi\,d\sigma, \text{ for all } \chi \in H^1(\Omega) \tag{19}$$

provided that $\mathrm{A}(x)$ satisfies (A1) and $\rho \in L_p(\Omega)$, for some $p > 1$. When $\rho = 0$, (19) is a NA-Harmonic equation, hence a unique solution requires the zero-mean condition.

The stream function ψ uniquely satisfies

$$\int_\Omega \{(\mathrm{A}^{-1}\mathrm{Curl}\,\psi)\cdot\mathrm{Curl}\chi - \omega\chi\}dx = 0, \text{ for all } \chi \in H_0^1(\Omega). \tag{20}$$

provided that $\mathrm{A}(x)$ satisfies (A1) and $\omega \in L_p(\Omega)$, for some $p > 1$.

The construction of the harmonic vector function h depends on the J "holes" in the geometry. When there are no holes, then $\mathrm{h}(\mathrm{x}) = 0$, see [2] for details. Otherwise,

$$h(x) = A(x)^{-1}\sum_{k=0}^{J} c_k \mathrm{Curl}\,\psi_k(x) \tag{21}$$

where each $\psi_k \in \{u \in H^1(\Omega) | u = 1\ on\ \gamma_j\ and\ \mathrm{u} = 0\ on\ \partial\Omega\backslash\gamma_j\}$ satisfies

$$\int_\Omega (\mathrm{A}^{-1}\mathrm{Curl}\ \psi_k)\cdot\mathrm{Curl}\chi\,dx = 0, \text{ for all } \chi \in H^1(\Omega). \tag{22}$$

The c_k coefficients can be resolved provided circulations $\int_{\gamma_j}\omega = \kappa_j(\omega)$ are prescribed on the holes for $1 \le j \le J$. Let $c = (c_1, c_2, \dots, c_J)$ be the coefficients,

K be a matrix with $k_{i,j} = \langle A(x)^{-1}\mathrm{Curl}\,\psi_i(x), A(x)^{-1}\mathrm{Curl}\,\psi_j(x)\rangle_A$ and $\kappa = (\kappa_1, \kappa_2, \dots, \kappa_J)$ then $c = -K^{-1}\kappa$.

(iv) **Steklov-Eigenfunction Expansion Method (SEM)**

SEM for the mixed DN problem seeks to find all $s \in H^1(\Omega)$ and $\delta \in \mathbb{R}$ such that

$$\int_\Omega (\mathrm{A}\nabla s) \cdot \nabla \mathrm{v}\, dx = \delta \int_\Sigma s\, \mathrm{v}\, d\sigma, \text{ for } v \in H^1(\Omega) \quad (23)$$

$$with \int_\Sigma |s|^2\, d\sigma = 1. \quad (24)$$

Substituting $v = s$ into (23) and applying (9) yields

$$\mathrm{c}_1 \int_\Omega |\nabla \mathrm{s}|^2 dx \geq \int_\Omega (\mathrm{A}\nabla s) \cdot \nabla \mathrm{s}\, dx = \delta \geq \mathrm{c}_0 \int_\Omega |\nabla \mathrm{s}|^2 dx.$$

$s \equiv$ constant entails $\delta = 0$, otherwise $\int_\Omega |\nabla \mathrm{s}|^2 dx > 0$, hence $\delta > 0$.

There are an infinite number of Steklov-eigenvalues

$$0 = \delta_0 < \delta_1 \leq \delta_2 \leq \cdots \leq \delta_k \leq \cdots \quad (25)$$

with associated Steklov-eigenfunctions $s_k(x)$ such that

$$[s_m(x)\,, s_n(x)\,]_{A\Sigma} = 0 \text{ and } \langle s_m(x)\,, s_n(x)\rangle_\Sigma = 0. \quad (26)$$

$\{s_k(x)\}$ forms a mutual orthonormal basis for $ker\mathcal{L}(\Sigma)$, see [3] for details.

Given $u \in H^1_\Sigma(\Omega)$, u can be represented by Steklov-eigenfunctions. The sum of Steklov-eigenfunctions converges strongly to u. This result is proven in [3].

$$u(x) = \sum_{k=1}^{\infty} \frac{[s_k\,, u\,]_{A\Sigma}}{[s_k\,, s_k\,]_{A\Sigma}} s_k(x) = \sum_{k=1}^{\infty} \langle s_k\,, u\rangle_\Sigma\, s_k(x) \quad (27)$$

For the NA-Harmonic problem in (19), let Σ^* denote the subset of the boundary $\partial\Omega$ with non-zero flux. Then for the NA-Harmonic equation with $\rho = 0$, SEM seeks to find all $s \in H^1(\Omega)$ and $\delta \in \mathbb{R}$ such that

$$\int_\Omega (\mathrm{A}\nabla s) \cdot \nabla \mathrm{v}\, dx = \delta \int_{\Sigma^*} s\, \mathrm{v}\, d\sigma, \text{ for } v \in H^1(\Omega) \quad (28)$$

$$with \int_{\Sigma^*} |s|^2\, d\sigma = 1. \quad (29)$$

Notice that (28) and (29) are similar to (25) and (26).

$\{s_k(x)\}$ forms a mutual orthonormal basis for a subspace of $\mathcal{H}_A(\Omega)$, the subspace of A-Harmonic functions of with vanishing flux on Σ^* in $H^1(\Omega)$.

Given $\in H^1(\Omega)$, let $\varphi(x) = \sum_{k=1}^{\infty} c_k\, s_k(x)$ be the Steklov-expansion. Substituting φ into (19) yields

$\sum_{k=1}^{\infty} c_k \int_{\Sigma^*} (\mathrm{A}\nabla s_k) \cdot \nabla \chi\, d\sigma = \sum_{k=1}^{\infty} c_k\, \delta_K \langle s_k, \chi\rangle_{\Sigma^*} = \int_{\Sigma^*} \mu\chi\, d\sigma = \langle \mu\,, \chi\rangle_{\Sigma^*}$, for some $\chi \in H^1(\Omega)$.

Let $\chi = s_i$ then by A-orthogonality, $c_i = \int_{\Sigma^*} \mu\chi\, d\sigma/\delta_i = \langle \mu\,, s_i\rangle_{\Sigma^*}/\delta_i$ yields

$$\varphi(x) = \sum_{k=1}^{\infty} \langle \mu\,, s_k\rangle_{\Sigma^*}\, s_k(x)/\delta_k. \quad (30)$$

(v) Variational form for FEM and SEM Representation for A-Harmonic Vector and Scalar Fields

A-Harmonic Scalar Field

In order for the mixed DN bvp (1)-(2) to have a unique solution, the following condition must be satisfied:

(A2) – The Dirichlet boundary data g must be in the trace space, $g \in H^{1/2}(\Sigma)$.

Let $H^1_{\Sigma g}(\Omega) \coloneqq \{\mathrm{u} \in \mathrm{H}^1(\Omega) | \Gamma u = g, on\ \Sigma\}$.
The variational form of (1)-(2) seeks to find a $u \in H^1_{\Sigma g}(\Omega)$ such that

$$\int_\Omega (\mathrm{A}\nabla u) \cdot \nabla \mathrm{v}\, dx = 0, \text{ for all } v \in H^1_{\Sigma 0}(\Omega)\,. \tag{31}$$

(31) is solved using traditional FEM.

Alternatively, the solution u has the Steklov-eigenfunction expansion given by (27) with g on the boundary

$$u(x) = \sum_{k=1}^{\infty} \langle s_k\,, g\rangle_\Sigma\, s_k(x). \tag{32}$$

A-Harmonic Vector Field

In order for the N div-curl sys (6)-(8) with constant divergence and constant curl to have a unique solution, the following compatibility condition must be satisfied:

(A3) – The prescribed flux data μ must satisfy $\int_{\partial\Omega} \mu d\sigma = \int_\Omega \rho dx$.

From (18), there is a *potential function* $\varphi \in H^1(\Omega)$, *stream function* $\psi \in H^1_0(\Omega)$ and *harmonic vector function* $\mathrm{h} \in \mathcal{H}_{Ev0}(\Omega)$ such that $u(x) = \nabla\varphi(x) + A(x)^{-1} Curl\ \psi(\mathrm{x}) + \mathrm{h}(\mathrm{x})$.
The potential function φ must satisfy

$$\int_\Omega (\mathrm{A}\nabla\varphi) \cdot \nabla\chi\ dx = \int_{\partial\Omega} \mu\chi\, d\sigma, \text{ for all } \chi \in H^1(\Omega). \tag{33}$$

The stream function ψ must satisfy

$$\int_\Omega (\mathrm{A}^{-1}\mathrm{Curl}\ \psi) \cdot \mathrm{Curl}\chi dx = 0, \text{ for all } \chi \in H^1_0(\Omega). \tag{34}$$

Hence, ψ must be a constant. $\psi \in H^1_0(\Omega)$ implies $\psi = 0$.

If we further assume the prescribed circulations $\int_{\gamma_j}\omega = 0$ then $\mathrm{h(x)} = 0$. Problem (6)-(8) is simplified to the following variational problem (NA-Harmonic problem).

Let $H^1_{\mathrm{m}}(\Omega) \coloneqq \{\mathrm{u} \in \mathrm{H}^1(\Omega) | \int_\Omega u(x)dx = 0\}$, the subspace of functions in $\mathrm{H}^1(\Omega)$ with zero mean.
Find $\varphi \in H^1_m(\Omega)$ such that

$$\int_\Omega (\mathrm{A}\nabla\varphi)\cdot\nabla\chi \; dx = \int_{\Sigma^*}\mu\chi \, d\sigma, \text{ for all } \chi \in H^1_{\mathrm{m}}(\Omega). \tag{35}$$

The vector field solution is then given by

$$u(x) = \nabla\varphi + Mx. \tag{36}$$

Alternatively, the solution φ has the Steklov-eigenfunction expansion given by (30)

$$\varphi(x) = \sum_{k=1}^{\infty} \langle \mu \, , s_k\rangle_{\Sigma^*}/\delta_k \, s_k(x) \tag{37}$$

The vector field then solution in terms of Steklov-eigenfunctions is then

$$u(x) = \sum_{k=1}^{\infty} \frac{\langle \mu \, , s_k\rangle_{\Sigma^*}}{\delta_k}\nabla s_k(x) + Mx. \tag{38}$$

Problem (35) generally is difficult to implement in FEM directly as it is non-trivial to create a finite element space that guarantee the mean of the function is zero. Instead, it is preferred to solve (35) using a mixed finite element method that enforces the constraint with a Lagrange multiplier. The mixed variational formulation is to find $\varphi \in H^1(\Omega)$ and $\lambda \in \mathbb{R}$ such that

$$\int_\Omega (\mathrm{A}\nabla\varphi)\cdot\nabla\chi \; dx + \int_\Omega \varphi\beta \; dx + \int_\Omega \lambda\chi \; dx = \int_{\partial\Omega}\mu\chi \, d\sigma, \text{ for all } \chi \in H^1(\Omega), \beta \in \mathbb{R}. \tag{39}$$

5. **Analytical Solution of Steklov-Eigenvalues on a Rectangle using SEM**

SEM problems (23)-(24) and (28)-(29) have solutions that can be calculated directly using the method of separation of variables for the case when the geometry is a rectangle. The classical problem is posed below on a rectangle that has a length 2*L* and width 2*M* such that $L \geq M$. This spectrum is useful for comparing numerical results in section 7 for the mixed DN bvp and Neumann bvp.

$$\Delta u(x,y) = 0, for\ (x,y) \in \Omega, \tag{SE.1}$$

$$\frac{\partial u(x,M)}{\partial y} = \frac{\partial u(x,-M)}{\partial y}, for\ x \in (-\mathrm{L},\mathrm{L}), \tag{SE.2}$$

$$-\frac{\partial u(-L,y)}{\partial x} = \delta u(-L,y), for\ y \in (-\mathrm{M},\mathrm{M}), \tag{SE.3}$$

$$\frac{\partial u(L,y)}{\partial x} = \delta u(L,y), for\ y \in (-\mathrm{M},\mathrm{M}). \tag{SE.4}$$

Applying separation of variables to (SE.1) yields $u(x,y) = w(x)v(y)$, for some real-valued functions w: $[-L,L] \to \mathbb{R}$ and v: $[-M,M] \to \mathbb{R}$ with non-empty support.

$$\Delta u(x,y) = \frac{\partial^2 w(x)}{\partial x^2} v(y) + \frac{\partial^2 v(y)}{\partial y^2} w(x) = 0, for\ (x,y) \in \Omega. \tag{SE.5}$$

Let $C = supp(w) \times supp(v)$, the subset in the rectangle Ω where the functions are non-zero,

$$\frac{1}{w(x)}\frac{\partial^2 w(x)}{\partial x^2} = -\frac{1}{v(y)}\frac{\partial^2 v(y)}{\partial y^2}, \qquad for\ (x,y) \in \mathrm{C}. \tag{SE.6}$$

Fix $x^* \in supp(w)$, then $-\frac{1}{v(y)}\frac{\partial^2 v(y)}{\partial y^2} = \lambda = \frac{1}{w(x^*)}\frac{\partial^2 w(x^*)}{\partial x^2}$ is a constant for all $y \in supp(v)$. Similarly, $\lambda = \frac{1}{w(x)}\frac{\partial^2 w(x)}{\partial x^2}$ is a constant for all $x \in supp(w)$.

$$\frac{\partial^2 w(x)}{\partial x^2} - \lambda\, w(x) = 0\ and\ \frac{\partial^2 v(y)}{\partial y^2} + \lambda\, v(y) = 0, v\ for\ (x,y) \in \mathrm{C}. \tag{SE.7}$$

For $x \in [-L,L] \backslash supp(w)$, $\frac{\partial^2 w(x)}{\partial x^2} v(y) = 0, for\ all\ y \in (-M,M)$ implies $\frac{\partial^2 w(x)}{\partial x^2} = 0$ otherwise $supp(v) = \emptyset$. Similarly, $y \in [-M,M] \backslash supp(v)$ implies $\frac{\partial^2 v(y)}{\partial y^2} = 0$. (SE.7) holds for the entire region.

$$\frac{\partial^2 w(x)}{\partial x^2} - \lambda\, w(x) = 0\ and\ \frac{\partial^2 v(y)}{\partial y^2} + \lambda\, v(y) = 0, for\ (x,y) \in \Omega. \tag{SE.8}$$

For $\lambda < 0$, there are no non-trivial solutions for (SE.8). Assume $\lambda < 0$ then $v(y) = \mathrm{c}_1 \cosh\sqrt{-\lambda}\, y + \mathrm{c}_2 \sinh\sqrt{-\lambda}\, y$, for some real-valued constants $\mathrm{c}_1, \mathrm{c}_2$. Applying the boundary conditions (SE.2) yields the linear system

$$\begin{pmatrix} -\sinh\sqrt{-\lambda}\, M & \cosh\sqrt{-\lambda}\, M \\ \sinh\sqrt{-\lambda}\, M & \cosh\sqrt{-\lambda}\, M \end{pmatrix} \begin{pmatrix} \mathrm{c}_1 \\ \mathrm{c}_2 \end{pmatrix} = 0 \qquad \text{(SE.9)}$$

which has the unique solution $\begin{pmatrix} \mathrm{c}_1 \\ \mathrm{c}_2 \end{pmatrix} = 0$ as the determinant $-2\sinh\sqrt{-\lambda}\, M \cosh\sqrt{-\lambda}\, M < 0$ for $\lambda < 0$.

For $\lambda = 0$ there are constants $\mathrm{c}_1, \mathrm{c}_2, d_1, \mathrm{d}_2$ such that

$$w(x) = \mathrm{c}_1 + \mathrm{c}_2 x \qquad \text{(SE.10)}$$
$$\mathrm{v(y)} = \mathrm{d}_1 + \mathrm{d}_2 y. \qquad \text{(SE.11)}$$

Applying the boundary conditions (SE.2) to (SE.11) yields the linear system

$$\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{d}_1 \\ \mathrm{d}_2 \end{pmatrix} = 0 \qquad \text{(SE.12)}$$

which has a non-trivial solution $\mathrm{d}_2 = 0$ and d_1 a real value. The solution is a constant.

$$\mathrm{v}_0(\mathrm{y}) = \mathrm{d} \qquad \text{(SE.13)}$$

Applying the boundary conditions (SE.3) and (SE.4) to (SE.10) for λ =0 yields the linear system

$$\begin{pmatrix} \delta & 1 - \delta L \\ \delta & L\delta - 1 \end{pmatrix} \begin{pmatrix} \mathrm{c}_1 \\ \mathrm{c}_2 \end{pmatrix} = 0 \qquad \text{(SE.14)}$$

In terms of δ there are two non-trivial solutions of (14) , namely $\delta_1^0 = 0$ and $\delta_2^0 = \frac{1}{L}$. When $\delta_1^0 = 0$, $w_1^0(x) = \mathrm{c}_1$ is a constant and $\delta_2^0 = \frac{1}{L}$ corresponds to $w_1^0(x) = \mathrm{c}_2 x$. Clearly, $\delta_1^0 \leq 1/L$. c_1 can be selected to enforce $w_1^0(x)$ to be a zero-mean function while c_2 can be chosen to normalize the eigenfunction with respect to the norm,$\|.\|_\Sigma = 1$.

For $\lambda > 0$ there are constants $\mathrm{c}_1, \mathrm{c}_2, d_1, \mathrm{d}_2$ such that

$$w(x) = \mathrm{c}_1 \cosh\sqrt{\lambda}\, x + \mathrm{c}_2 \sinh\sqrt{\lambda}\, x \qquad \text{(SE.15)}$$
$$\mathrm{v(y)} = \mathrm{d}_1 \cos\sqrt{\lambda}\, y + \mathrm{d}_2 \sin\sqrt{\lambda}\, y. \qquad \text{(SE.16)}$$

Applying the boundary conditions (SE.2) to (SE.16) yields the linear system

$$\begin{pmatrix} -\sin\sqrt{\lambda}\, M & \cos\sqrt{\lambda}\, M \\ \sin\sqrt{\lambda}\, M & \cos\sqrt{\lambda}\, M \end{pmatrix} \begin{pmatrix} \mathrm{d}_1 \\ \mathrm{d}_2 \end{pmatrix} = 0 \qquad \text{(SE.17)}$$

which has non-trivial solutions given by the determinant $-2\sin\sqrt{\lambda}\, M \cos\sqrt{\lambda}\, M = -\sin 2\sqrt{\lambda}\, M = 0$ $\sin 2\sqrt{\lambda}\, M = 0$ implies $\lambda_k = \frac{\pi^2 k^2}{4M^2}$, for $k \geq 1$. Let $\hat{\lambda}_k = \sqrt{\lambda_k} = \frac{\pi k}{2M}$. For even k, $\sin\hat{\lambda}_k M = 0$ and $\cos\hat{\lambda}_k M \neq 0$ and for odd k, $\sin\hat{\lambda}_k M \neq 0$ and $\cos\hat{\lambda}_k M = 0$.

There is a constant d such that

$$\mathrm{v(y)} = \frac{\mathrm{d}}{2}((1+(-1)^k)\, cos\hat{\lambda}_k\, y + (1-(-1)^k)\sin\hat{\lambda}_k\, y).$$

Observing the fact that $(-1)^k = \cos\pi\, k$ for integer *k* and, applying angle sum and sum-to-product trigonometric identities successively yields

$$\mathrm{v}_k(\mathrm{y}) = \mathrm{d}(-1)^{\lfloor k/2 \rfloor}\sin\frac{\pi}{2}(k\left(\frac{y}{M}-1\right)+1)\,, for\ k \geq 0. \qquad \text{(SE.18)}$$

Applying the boundary conditions (SE.3) and (SE.4) to (SE.15) for λ >0 yields the linear system

$$\begin{pmatrix} \sinh\sqrt{\lambda}\,L - \delta/\sqrt{\lambda}\cosh\sqrt{\lambda}\,L & \delta/\sqrt{\lambda}\sinh\sqrt{\lambda}\,L - \cosh\sqrt{\lambda}\,L \\ \sinh\sqrt{\lambda}\,L - \delta/\sqrt{\lambda}\cosh\sqrt{\lambda}\,L & \cosh\sqrt{\lambda}\,L - \delta/\sqrt{\lambda}\sinh\sqrt{\lambda}\,L \end{pmatrix}\begin{pmatrix} \mathrm{c}_1 \\ \mathrm{c}_2 \end{pmatrix} = 0. \qquad \text{(SE.19)}$$

In terms of δ there are two non-trivial solutions of (14) , namely $\delta_1 = \sqrt{\lambda}\, tanh\sqrt{\lambda}L$ and $\delta_2 = \sqrt{\lambda}\, coth\sqrt{\lambda}L$, which are well-defined for $\lambda > 0$.

Let $\delta_1^k = \hat{\lambda}_k tanh\,\hat{\lambda}_k L$ and $\delta_2^k = \hat{\lambda}_k coth\,\hat{\lambda}_k L$, for $k >= 1$. For all $k \geq 0$, $\delta_1^k \leq \delta_2^k \leq \delta_1^{k+1}$.
To see this, for $x > 0$, $0 < 1 = cosh^2 x - sinh^2 x \rightarrow sinh^2 x < cosh^2 x \rightarrow \tanh x < coth\, x$.
For $k \geq 1$, $\delta_1^k = \hat{\lambda}_k tanh\,\hat{\lambda}_k L \leq \hat{\lambda}_k\, coth\,\hat{\lambda}_k L = \delta_2^k$.
For $k \geq 1$, $\frac{coth\,\hat{\lambda}_k L}{tanh\hat{\lambda}_{k+1} L} < \frac{coth\hat{\lambda}_k L}{tanh\hat{\lambda}_k L} = 1 \leq \sqrt{1+\frac{1}{k}} = \hat{\lambda}_{k+1}/\hat{\lambda}_k$ .Thus, $\delta_2^k \leq \delta_1^{k+1}$.
Furthermore, $\delta_2^0 = \frac{1}{L} \leq \delta_1^1 = \hat{\lambda}_1 tanh\,\hat{\lambda}_1 L$.
To see this, $x \tanh x > 1$ when $x > 1.2$. $\hat{\lambda}_1 L > 1.2$ when $L >= M$.
Thus, for all $k \geq 0$, $\delta_1^k \leq \delta_2^k \leq \delta_1^{k+1}$.

For $k \geq 1$, δ_1^k corresponds to $\mathrm{w}_{\delta_1^k}(x) = \mathrm{c}_1 \cosh\hat{\lambda}_k\, x$ while δ_2^k corresponds to $\mathrm{w}_{\delta_2^k}(x) = \mathrm{c}_2 \sinh\hat{\lambda}_k\, x$.
For each integer $k \geq 0$, there are two solutions $\mathrm{w}_{\delta_1^k}, \mathrm{w}_{\delta_2^k}$. This sequence of eigenvalues and eigenfunctions can be arranged into sequences in such a way that the eigenvalue sequence is an increasing sequence for $k >= 1$.
Let $k = \left\lfloor \frac{n}{2} \right\rfloor$, for $n \geq 2$. The sequence of eigenfunctions are given by

$$\mathrm{w}_n(x) = c((1+(-1)^n)\cosh\frac{\pi\left\lfloor \frac{n}{2} \right\rfloor}{2M}x + (1-(-1)^n)\sinh\frac{\pi\left\lfloor \frac{n}{2} \right\rfloor}{2M}x).$$

This expression can be simplified further by applying $\cosh x + \sinh x = e^x$ and $\cosh x - \sinh x = e^{-x}$.

$$\mathrm{w}_n(x) = c(e^{\frac{\pi\left\lfloor \frac{n}{2} \right\rfloor}{2M}x} + (-1)^n e^{-\frac{\pi\left\lfloor \frac{n}{2} \right\rfloor}{2M}x}). \qquad \text{(SE.20)}$$

Combining (SE.13) and (SE.16), $u_n(x,y) = C\ \mathrm{w}_n(x)\mathrm{v}_{\left\lfloor \frac{n}{2} \right\rfloor}(\mathrm{y})$, then

$$u_n(x,y) = C(e^{\frac{\pi\left\lfloor \frac{n}{2} \right\rfloor}{2M}x} + (-1)^n e^{-\frac{\pi\left\lfloor \frac{n}{2} \right\rfloor}{2M}x})(-1)^{\left\lfloor \left\lfloor \frac{n}{2} \right\rfloor/2 \right\rfloor}\sin\frac{\pi}{2}\left(\left\lfloor \frac{n}{2} \right\rfloor\left(\frac{y}{M}-1\right)+1\right), for\ n \geq 2. \qquad \text{(SE.21)}$$

(SE.16) can be normalized with respect to the norm, $\|.\|_\Sigma = 1$.

6. **Numerical FEM**

The finite element method is employed to resolve the Steklov-eigenvalue problem and other variational problems. A minimal description of the classical Lagrange finite element method is given.

(i) Lagrange elements FEM Basics

This Lagrange finite element description is taken from Glowinski [6]. Henceforth, the region Ω is a polygonal domain of $\mathbb{R}^2$ (polyhedral domain of $\mathbb{R}^3$). Polygonal (polyhedral) domains can approximate general regions Ω with smooth boundaries and are referred to as a *mesh*. A triangulation (tetrahedralization) $\mathcal{T}_h$ of Ω is a finite subset of triangles (tetrahedron) T such that

$$T \subset \overline{\Omega}, \text{ for all}, T \in \mathcal{T}_h, \quad \cup_{T\in\mathcal{T}_h} T = \overline{\Omega} \tag{40}$$

$$\dot{T_1} \cap \dot{T_2} = \emptyset, \text{ for all}, T_1, T_2 \in \mathcal{T}_h, \text{with } T_1 \neq T_2. \tag{41}$$

For $T_1, T_2 \in \mathcal{T}_h$, with $T_1 \neq T_2$, exactly one of the conditions must hold

$$\begin{aligned} &T_1 \cap T_2 = \emptyset, \\ &T_1 \text{and } T_2 \text{ share a common vertex,} \\ &T_1 \text{and } T_2 \text{ share a common edge,} \\ &T_1 \text{and } T_2 \text{ share a common face (3d case).} \end{aligned} \tag{42}$$

The parameter h estimates the worst triangle size $h(\mathcal{T}_h) \coloneqq \max\{diam(T_k)|T_k \in \mathcal{T}_h\}$. The triangulation (tetrahedralization) is *regular* provided that

$$lim_{h\downarrow 0}(\mathcal{T}_h) = 0 \text{ and}$$

there is a number $\sigma > 0$ independent of h such that

$$\frac{\rho(T_k)}{diam(T_k)} \geq \sigma, \text{ for all } T_k \in \mathcal{T}_h$$

where $\rho(T_k)$ is the diameter of the inscribed circle of T_k. All triangulations are assumed to be regular.

For this work, only Lagrange finite element constructions are considered. A Lagrange kth polynomial space is defined in two dimensions as

$$P_k(\mathrm{S}) \coloneqq \{w|w(x,y) = \textstyle\sum_{0\leq i+j\leq k} c_{ij}x^i y^j, (x,y) \in S, c_{ij} \in \mathbb{R}\}, \tag{43}$$

and in three dimensions as

$$P_k(\mathrm{S}) \coloneqq \{w|w(x,y,z) = \textstyle\sum_{0\leq i+j+l\leq k} c_{ijl}x^i y^j z^l, (x,y,z) \in S, c_{ijl} \in \mathbb{R}\}. \tag{44}$$

The *conforming* finite element space is a subspace $V_h^k(\Omega) \subset V(\Omega) = H_\Sigma^1(\Omega)$ such that

$$V_{h,\Sigma\mathrm{g}}^k(\Omega) \coloneqq \{v \in V(\Omega)|\mathrm{v}|_T \in P_k(T) \; and \; \mathrm{v}|_e = g, if \; e \; is \; an \; edge \; of \; T \; on \; \Sigma, \forall \; T \in \mathcal{T}_h\} \tag{45}$$

where $v|_T \in P_k(T)$ denotes the function restriction of v to the triangle T as a member Lagrange k-th order polynomial space defined on triangle T. For triangles (tetrahedrons) with an edge (face) that meet the Dirichlet boundary Σ, special care must be taken to enforce this type of boundary condition. The dimension of the finite element subspace, $\dim V_h^k$, is equal to number of degrees of freedom which depends on the order *k* of $P_k(T)$. For this work, primarily P_1 and P_2 are used.

The subspace (45) can be used to convert the variational formulations for the Steklov Eigenfunctions (23)- (24) and (28)-(29) into a discrete eigenvalue problem $Kv = \delta Mv$, the mixed DN bvp (31) into a discrete linear system K$v = Mb$, and the N-A Harmonic bvp (39) described in section (v) into a linear block matrix system, $\begin{pmatrix} K & B \\ B^T & 0 \end{pmatrix}\begin{pmatrix} v \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$. K is referred to as a stiffness matrix, M as the mass matrix, B as the constraint matrix, b as the load vector, and v are the unknown degrees of freedom that are used to construct the solution.

(ii) Summarized Finite Element Approximations

SEM for the mixed DN bvp (23)-(24) seeks to find all $s \in V_h^k(\Omega)$ and $\delta \in \mathbb{R}$ such that

$$\int_\Omega (\mathrm{A}\nabla s) \cdot \nabla \mathrm{v}\, dx = \delta \int_\Sigma s\, \mathrm{v}\, d\sigma, \text{ for } v \in V_h^k(\Omega) \quad (46)$$

$$with \int_\Sigma |s|^2\, d\sigma = 1. \quad (47)$$

SEM for the N-Harmonic bvp (28)-(29) seeks to find all $s \in V_h^k(\Omega)$ and $\delta \in \mathbb{R}$ such that

$$\int_\Omega (\mathrm{A}\nabla s) \cdot \nabla \mathrm{v}\, dx = \delta \int_{\Sigma^*} s\, \mathrm{v}\, d\sigma, \text{ for } v \in V_h^k(\Omega) \quad (48)$$

$$with \int_{\Sigma^*} |s|^2\, d\sigma = 1. \quad (49)$$

FEM for the mixed DN bvp (31) seeks to find $u \in V_{h,\Sigma \mathrm{g}}^k(\Omega)$ such that

$$\int_\Omega (\mathrm{A}\nabla u) \cdot \nabla \mathrm{v}\, dx = 0, \text{ for all } v \in V_{h,\Sigma 0}^k(\Omega). \quad (50)$$

Mixed FEM for NA-Harmonic bvp (39) seeks to find $\varphi \in V_h^k(\Omega)$ and $\lambda \in \mathbb{R}$ such that

$$\int_\Omega (\mathrm{A}\nabla \varphi) \cdot \nabla \chi\ dx + \int_\Omega \varphi\beta\, dx + \int_\Omega \lambda\chi\, dx = \int_{\partial\Omega} \mu\chi\, d\sigma, \text{ for all } \chi \in V_h^k(\Omega) \text{ and } \beta \in \mathbb{R}. \quad (51)$$

7. **FreeFEM++**

FreeFEM++ is a powerful numerical PDE solver developed by Friedrich Hecht at the Laboratoire Jacques-Louis Lions, Universite' Pierre et Marie Curie, Paris. FreeFEM++ is a compiler that operates on a unique programming language that derives several conventions (particularly computational mathematics) from Fortran, Matlab, and c++ while adding a powerful new variational framework that encapsulates the finite element methodology. It is an ideal tool for quickly prototyping numerical solutions to variational problems. For most problems in PDEs, the following work-flow can be conducted in FreeFEM++. A triangulated (tetrahedralized) geometry in the form of a mesh is either constructed or loaded into the system. This mesh can be either two-dimensional or three-dimensional depending on the nature of the problem. Next a finite element space or a collection of finite element spaces are constructed over the mesh. Each finite element space can have its own unique finite element type. Many common finite elements have been included in FreeFEM++; Lagrange elements and Thomas-Raviart elements are included, see [1] for more examples. Next a variational problem or a system of variational problems is declared over the finite element space(s). For some problems, it is enough to minimally specify the variational problem and immediately solve the problem using the high-level **solve** function provided in FreeFEM++. The solution can be post-processed and displayed in numerous ways. Some options include a traditional function plot, vector field plot, contour plot, and stream-line plot.

For problems that do not fit into the high-level solver mold, the variational problem can either be solved through optimization techniques or can be linearized into matrices. For eigenvalue problems, this is the only approach offered. There is a sufficient amount of numerical linear algebra routines that can uniquely resolve linear equations. If the variational problem of interest is an eigenvalue problem, there is a specialized spectral shift eigenvalue method built in FreeFEM++. In addition, FreeFEM++ is also capable of adapting the mesh, interpolating functions in finite element spaces, integrating and differentiating functions over the mesh and interacting with the operating system. This section demonstrates how to solve the mixed DN bvp and div-curl bvp in FreeFEM++.

(i) Meshing

The starting point for solving PDEs is to first construct a mesh that approximates the geometry of the problem. The example in this FreeFEM++ tutorial assumes a simple two dimensional rectangle of length $2 * L$ and width $2 * M$. For two-dimensional problems, the perimeter of the geometry must be specified. This is done using the **border** declaration. A border creates a parameterized curve. A collection of parameterized curves (borders) are then combined to form a closed region in the plane. If the closed region is valid, then a mesh can be created. The construction of three-dimensional geometry requires a different set of techniques and will not be discussed here.

The following code demonstrates the construction a rectangle and plots the rectangular mesh.

```
real      L   = 4.0;
real      M = 1.0;
border Left(t=- M, M)  {x=- L          ;  y=-t;   label=1;}    // Left wall
border Top(t=0, L)       {x= L -2.0*t ; y= M;   label=2;}  // Top  wall
border Right(t=- M, M){x=L;              y=t;     label=3;}   // Right wall
border Bottom(t=- L, L){x=t;             y=- M; label=4;}  // Bottom wall

// Construct a mesh
mesh Th = buildmesh(Left(5* M) + Top(5*L) + Right(5* M) + Bottom(5*L));
plot(Th);
```

Code Snippet 1: Demonstrates how to define the perimeter of a 2Mx2L rectangle, triangulate into a mesh and plot the mesh in FreeFEM++.

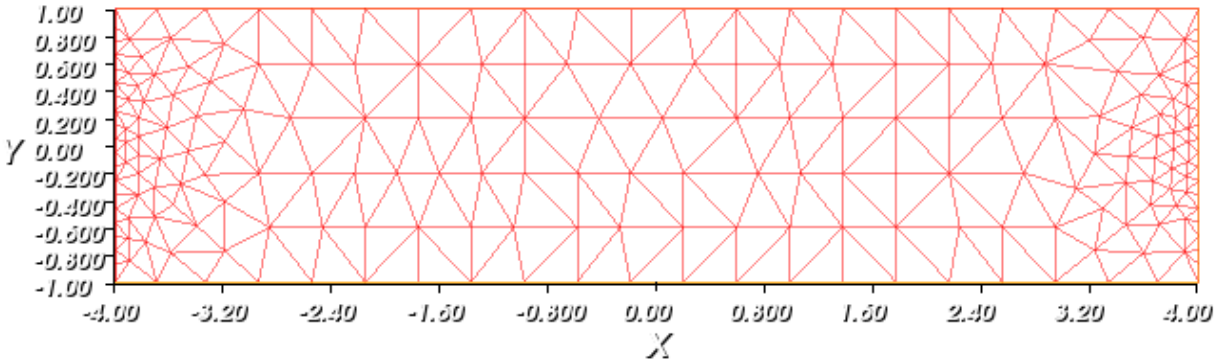


Figure 1: Plot of mesh from Code Snippet 1.

(ii) Defining a Finite Element Space

A finite element space is constructed over the mesh T_h. The following code sample declares a simple P1 Lagrange finite element space, defines a continuous function g, approximates g in this space and plots the approximation of g.

```
// Create P1 Lagrange Finite Element function space
fespace Vh(Th,P1);

// Define a function
func g=(x/L)^2+(y/M)^2;

// Approximate the function in Vh
Vh   uh = g;
plot(uh);
```

Code Snippet 2: Demonstrates how to declare a function space and function g in FreeFEM++. The function g is then approximated in this function space and plotted.

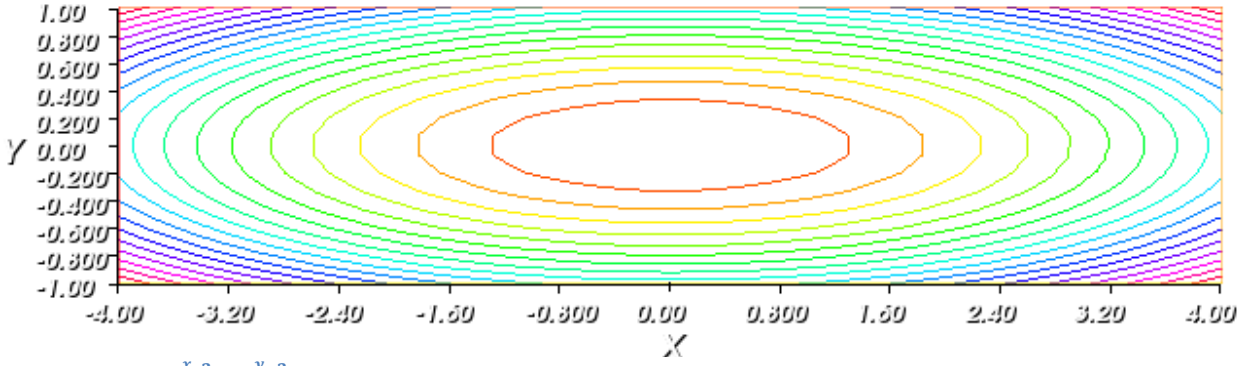


Figure 2: Plot of $(\frac{x}{L})^2 + (\frac{y}{M})^2$ from Code Snippet 2.

(iii) Solving a PDE Problem as a Variational Problem using Solve

The next snippet of code solves the mixed DN bvp (50) over the rectangle. It is a well-posed classical problem and fits directly into the FreeFEM++ language.

A recap of the problem statement: Find $u \in H^1_{\Sigma_g}(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla \mathrm{v}\, dx = 0, \text{ for all } v \in H^1_{\Sigma_0}(\Omega)\,.$$

Specifically, the Dirichlet boundary condition for demonstration is given by

$$u(-L,y) = g_1(y) = 1 - (\tfrac{y}{M})^2 \text{ and } u(L,y) = g_2(y) = y * (1 - \left(\tfrac{y}{M}\right)^2), \text{for } -M \le y \le M.$$

A homogenous Neumann condition is enforced on the remainder of the boundary. This is already handled by the variational definition above. The **solve** statement in FreeFEM++ is used to solve the variational problem. The variational problem here is in terms of a two-dimensional integral. The

integral is specified using the **int2d** keyword and includes the mesh T_h as a parameter. The Dirichlet conditions are enforced using the **on** keyword with the corresponding perimeter label found in the perimeter definition.

```
// Create P2 Lagrange Finite Element function space (Higher polynomial approximation)
fespace Vh(Th,P2);
Vh uh,vh;                  // uh wil be the solution function, vh is the trial function

// Build Dirichlet boundary condition functions
func g1=1-(y/M)^2;
func g2=y*(1-(y/M)^2);

// Mixed BVP (31) to solve
solve MixedBVP(uh,vh,solver=sparsesolver) =
        int2d(Th)(dx(uh)*dx(vh)+dy(uh)*dy(vh)) + on(1,uh=g1) + on(3,uh=g2);

plot(uh, fill=true, value=true); // Plot the solution with contours filled in and a legend visible
```

Code Snippet 3: Demonstrates how to directly solve a well-posed mixed BVP problem with Dirichlet conditions on the left and right walls, and homogenous Neumann conditions on the top and bottom walls.

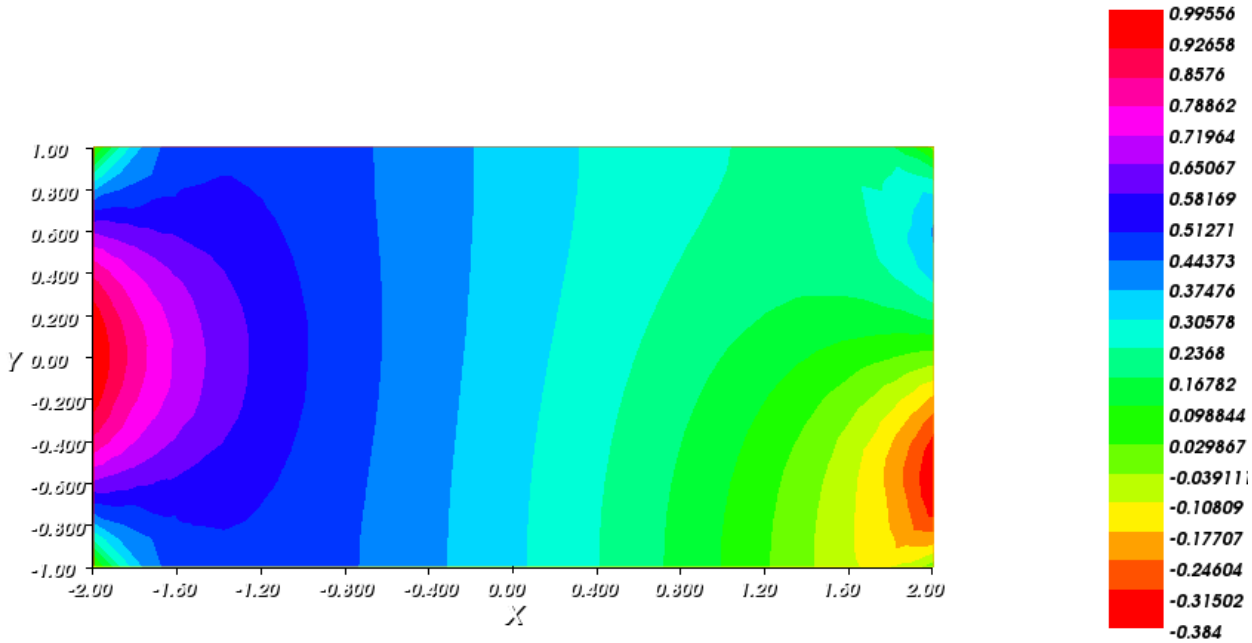


Figure 3: Plot of solution to mixed BVP produced in Code Snippet 3.

(iv) Solving a PDE Problem as a Variational Problem Explicitly

The next snippet of code solves the NA-Harmonic bvp (51) over the rectangle. This problem seeks solutions that have zero mean and the solve operator does not support this feature

A recap of the problem statement: Find $\varphi \in V_h^k(\Omega)$ and $\lambda \in \mathbb{R}$ such that

$$\int_\Omega \nabla\varphi \cdot \nabla\chi \; dx + \int_\Omega \varphi\beta \; dx + \int_\Omega \lambda\chi \; dx = \int_{\partial\Omega} \mu\chi \; d\sigma, \text{ for all } \chi \in V_h^k(\Omega) \text{ and } \beta \in \mathbb{R}.$$

Specifically, an inhomogeneous Neumann boundary condition is given on part of the boundary by

$$-\frac{\partial u}{\partial x}(-L, y) = g_1(y) = 1 - (\frac{y}{M})^2 \text{ and } \frac{\partial u}{\partial x}(L, y) = g_2(y) = (1 - \left(\frac{y}{M}\right)^2), \text{ for } -M \le y \le M.$$

A homogenous Neumann boundary condition applies to the rest of the boundary. A block matrix and vector is constructed that embeds the zero-mean constraint and variational problem. This system is solved and the solution vector contains the FEM solution and the Lagrange multiplier factor.

```
fespace Vh (Th,P1);  // Create a P1-Lagrange FEM space
func g1=1-(y/M)^2;   // Build Neumann boundary condition functions
func g2=1-(y/M)^2;
Vh uh, vh;               // Instantiate instances of finite element space

varf va(uh, vh) = int2d(Th)( dx(uh)*dx(vh)+dy(uh)*dy(vh))    // Stiffness part
varf vb(uh, vh) = int2d(Th)(vh);                             // Zero-mean
varf vL(uh, vh) = -int1d(Th,1)(g1* vh) +                     // Left wall
                   int1d(Th,3)(g2* vh);                      // Right wall

// Capture # of degrees of freedom to construct block matrix
int n1 = Vh.ndof +1;
// Construct a block matrix to solve mixed FEM problem
//   A*uh + B*l = b
//   B*uh=0
matrix   A = va(Vh, Vh);    // Stiffness Block matrix A
real[int] b = vL(0, Vh);      // Solution vector
real[int] B = vb(0, Vh);      // Constraint block matrix
real[int] bb(n1),             // Modified solution vector for mixed problem
          xx(n1),             // Solution vector
          l(1);               // Lagrange multiplier
matrix AA = [[A,B],[B',0]];     // Mixed block matrix
set(AA,solver=sparsesolver); // Set solver type on matrix
bb = [b,0];                      // Construct right hand side vector
xx = (AA^-1)*bb;                 // Get solution vector
[uh[],l] = xx;                   // Part of vector has solution, other has Lagrange multiplier
plot(uh,fill=true,value=true);
```

Code Snippet 4: Demonstrates how to solve a Neumann BVP problem with inhomogeneous conditions on the left and right walls, and homogenous Neumann conditions on the top and bottom walls.

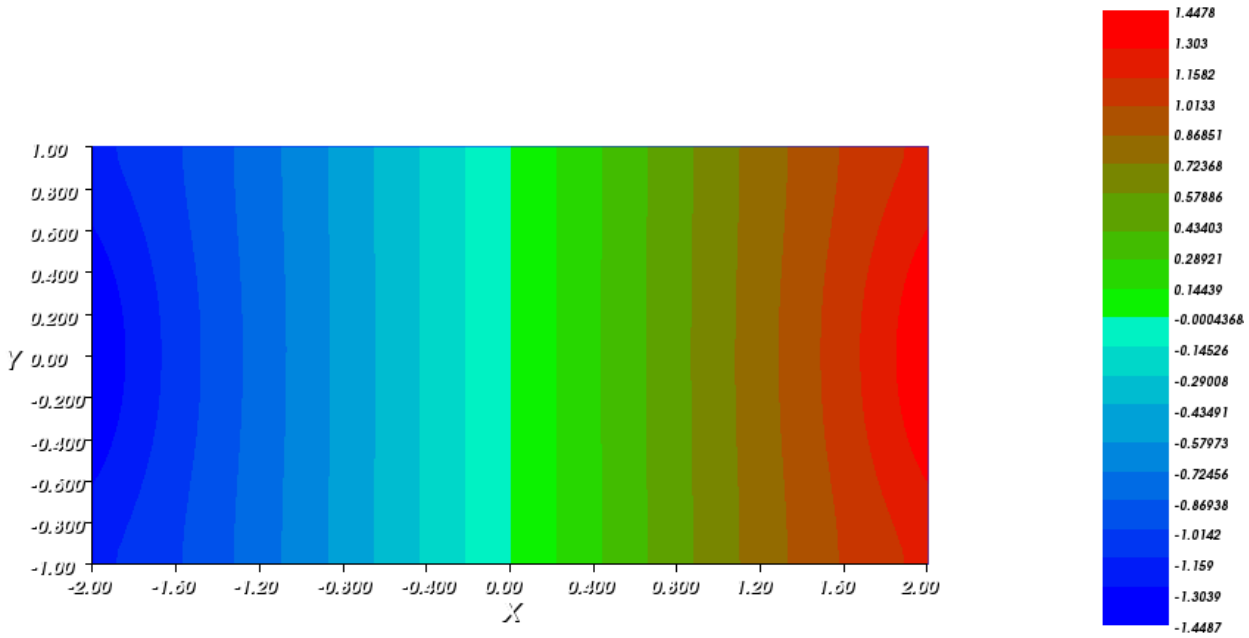


Figure 4: Solution to the Neumann bvp from Code Snippet 4.

The gradient of the solution can also be calculated and plotted in FreeFEM++ with the following code.

```
Vh xh = dx(uh),
    yh = dy(uh);    // Take derivatives of the solution in the x and y direction
plot([xh,yh]);      // Plot the vector field
```

Code Snippet 5: Demonstrates how to calculate the gradient of a scalar function and plot it.

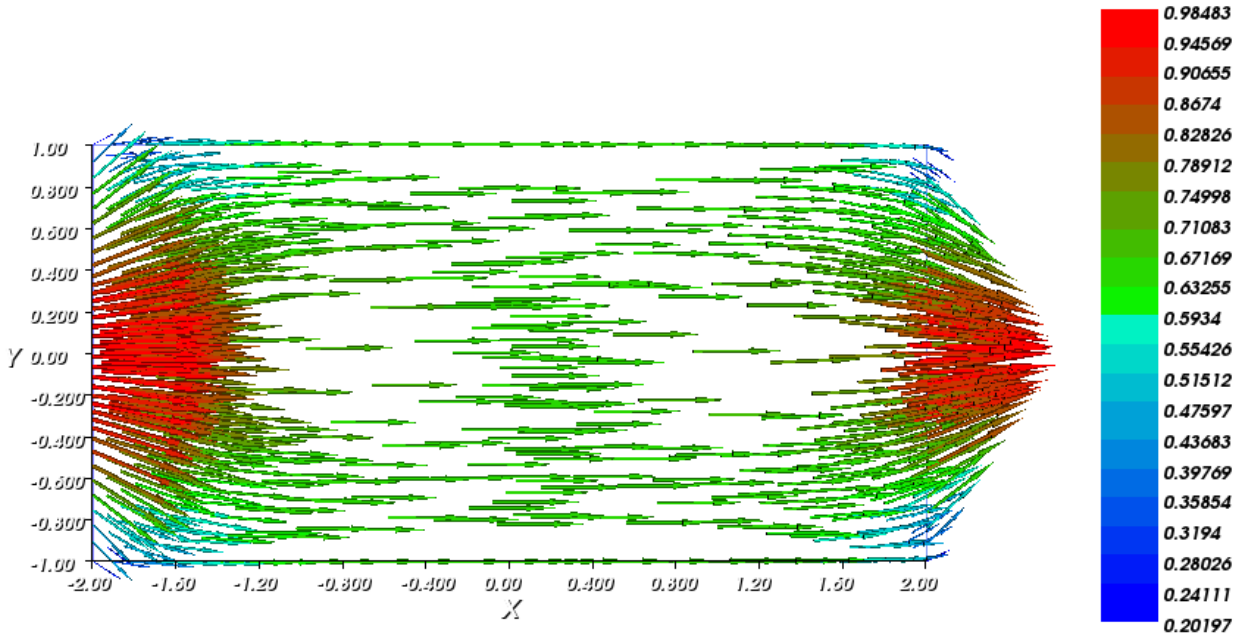


Figure 5: Plot of the gradient of a solution from Code Snippet 4 as a vector field.

(v) Solving an Eigenvalue Problem as a Variational Problem Explicitly

Similar to the Neumann bvp, the Steklov eigenvalue problems (46)-(49) must be solved at the matrix level. First, the **varf** keyword is used to define the variational forms in (46)-(49). This is used to construct two matrices, A and B. These matrices are then sent to the **EigenValue** routine which uses *Arpack* (a spectral shift method developed at Rice University) to solve eigenvalue problems. It is important that Dirichlet conditions only be specified on the variational forms that construct matrix A and not on the variational forms that construct matrix B otherwise an ill-conditioned system is created.

SEM for the mixed DN bvp and NA harmonic bvp (identifying the fact that $\Sigma = \Sigma^*$ for this demonstration) seeks to find all $s \in V_h^k(\Omega)$ and $\delta \in \mathbb{R}$ such that

$$\int_\Omega (\mathrm{A}\nabla s) \cdot \nabla \mathrm{v}\, dx = \delta \int_\Sigma s\, \mathrm{v}\, d\sigma, \text{ for } v \in V_h^k(\Omega)$$

$$with \int_\Sigma |s|^2\, d\sigma = 1.$$

The code below solves the Steklov-eigenvalue problem used to reconstruct solutions to both the mixed DN bvp and NA harmonic bvp.

```
fespace Vh (Th,P1);  // Create a P1-Lagrange FEM space
Vh uh, vh;               // Instantiate instances of finite element space

// Create variational form for Steklov-Eigenvalue problem
varf va(uh, vh) = int2d(Th)( dx(uh)*dx(vh)+dy(uh)*dy(vh));
varf vb(uh, vh) = int1d(Th,1,3)( uh * vh);

// Construct matrices to solve eigenvalue problem
//   A*x=l*B*x
matrix A = va(Vh, Vh ,solver = sparsesolver);    // Matrix A on left hand side
matrix B = vb(Vh, Vh);                           // Matrix B on right hand side

int eigCount = 6;          // Get first 6 Eigenvalues
real[int] ev(eigCount);  // Holds Eigenfunctions
Vh[int] eV(eigCount);        // Holds Eigenvalues

// Solve Ax=lBx
int numEigs = EigenValue(A,B,sym=true,sigma=0,value=ev,vector=eV);
numEigs = min(eigCount,numEigs);

for(int i=0;i<numEigs;i++)  // Plot the spectrum
   plot(eV[i],fill=true,value=true,cmm= ev[i]);
```

Code Snippet 6: Demonstrates how to solve a Steklov-eigenvalue problem. The first six eigenfunctions are plotted.

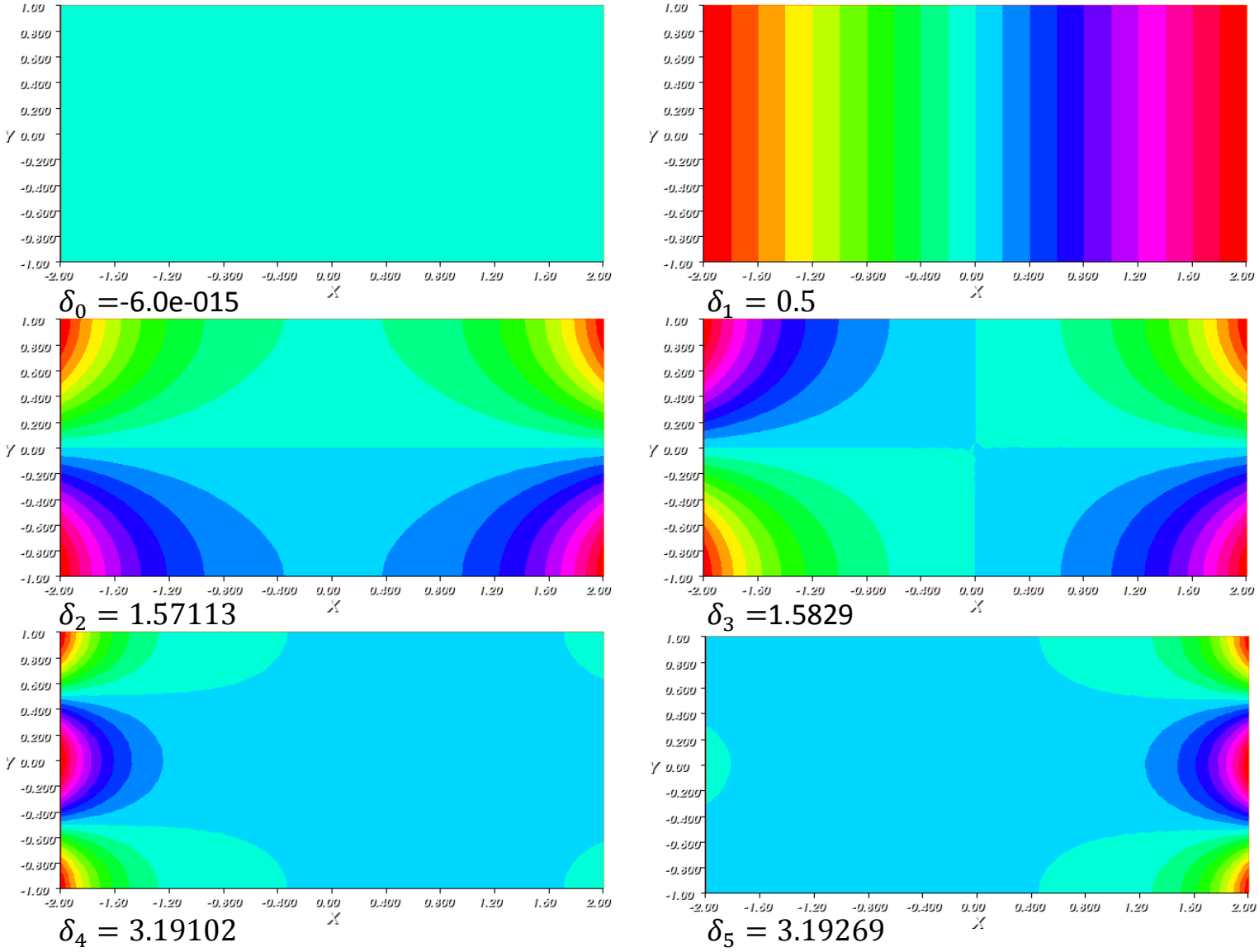


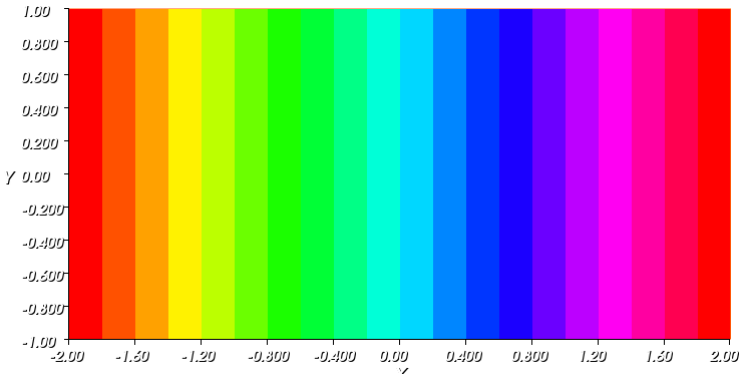
Figure 6: The first six Steklov-eigenfunctions from Code Snippet 6. The functions are displayed left-to-right in increasing eigenvalue order. The first eigenvalue is zero and corresponds to a constant function.

(vi) Solution Reconstruction via Steklov-Eigenfunctions

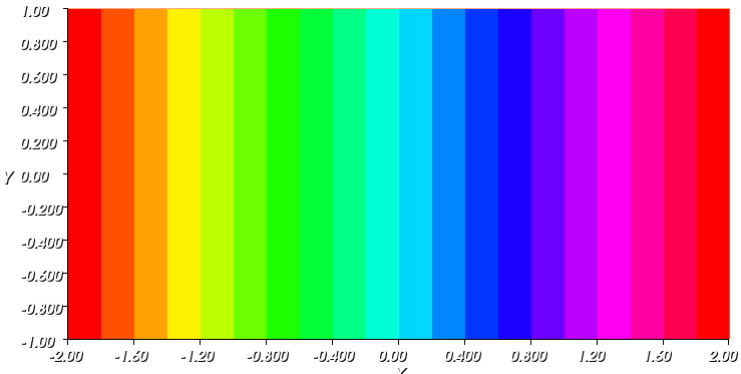
The Steklov-eigenfunctions can be used to approximate the solutions of the mixed DN bvp and NA harmonic bvp. After the eigenfunctions have been calculated, the coefficients in (32) and (38) can be calculated by evaluating boundary integrals that involve the prescribed data. The following code constructs approximations to the mixed DN bvp and NA harmonic bvp with the 5 non-zero eigenfunctions displayed in Figure 6. The partial summation of Steklov-terms is plotted. The final summation can be compared with the solutions found by FEM (mixed FEM) in subsections (iii) and (iv).

```
Vh uhDN = 0,   // Solution to reconstruct
    uhNA = 0;

// Build Neumann boundary condition functions
func g1=1-(y/M)^2;
func g2=1-(y/M)^2;

for(int i=1;i<numEigs;i++)
{
    real c =(-int1d(Th,1)(g1*eV[i]) + int1d(Th,3)(g2*eV[i]))/ ev[i];
    uhDN = uhDN + c * eV[i];
    plot(uhDN,fill=true);
}

// Build Dirichlet boundary condition functions
func z1=1-(y/M)^2;
func z2=y*(1-(y/M)^2);

for(int i=1;i<numEigs;i++)
{
    real c =(int1d(Th,1)(z1*eV[i]) + int1d(Th,3)(z2*eV[i]));
    uhNA = uhNA + c * eV[i];
    plot(uhNA,fill=true);
}
```

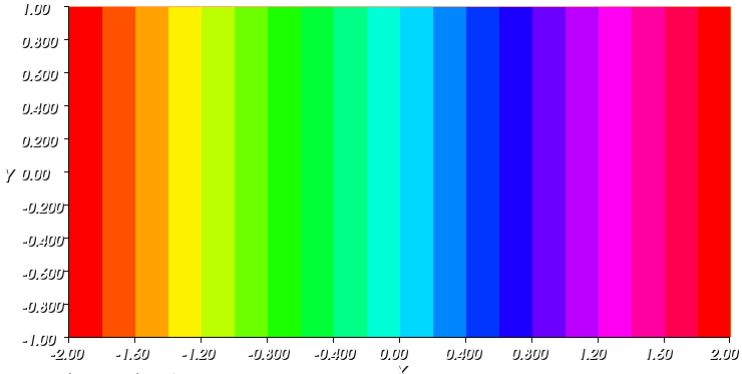
Code Snippet 7: Demonstrates how to use the Steklov-eigenfunctions to construct solutions to the mixed DN bvp and NA harmonic bvp. Partial summation with Steklov-eigenfunctions are plotted.



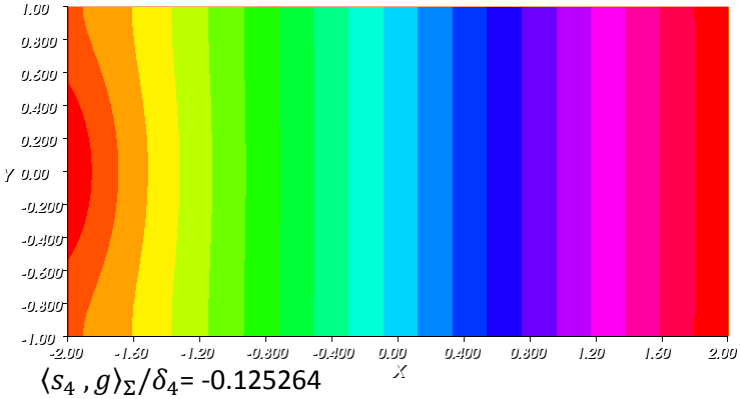
$\langle s_1 , g\rangle_\Sigma/\delta_1$ = 2.66667



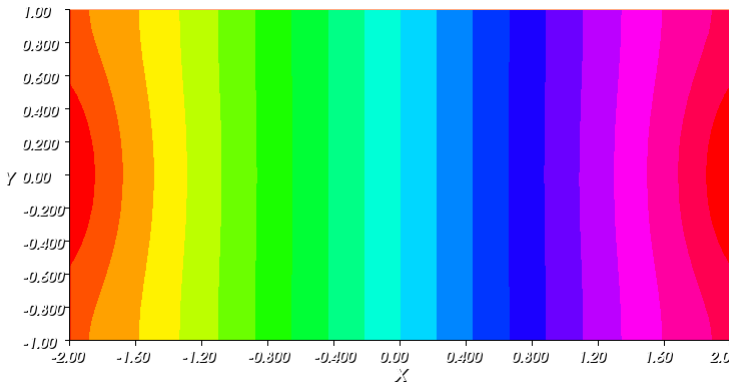
$\langle s_2 , g\rangle_\Sigma/\delta_2$= 2.92e-005



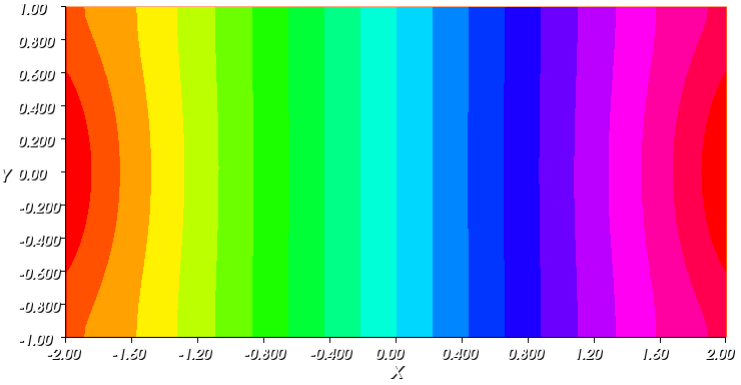
$\langle s_3 , g\rangle_\Sigma/\delta_3$= 2.22619e-005



$\langle s_4 , g\rangle_\Sigma/\delta_4$= -0.125264

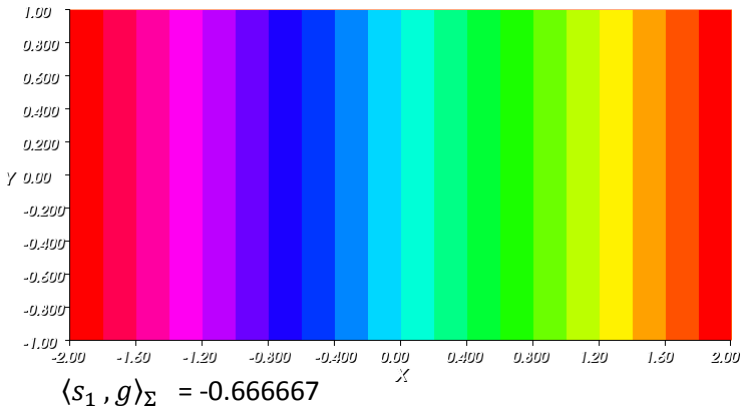


$\langle s_5 , g\rangle_\Sigma/\delta_5$= 0.128671

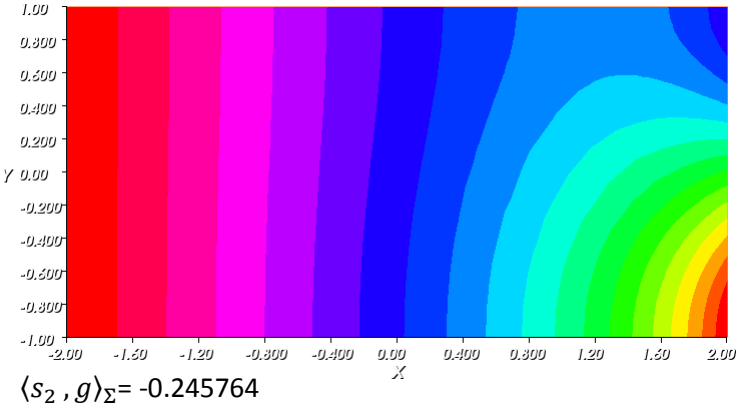


mixed FEM solution

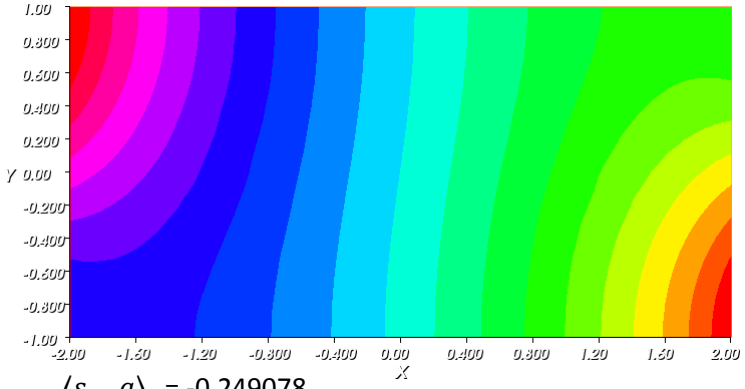
Figure 7: The plots show the reconstruction of the solution to the NA harmonic bvp. Each plot contains a partial sum of a certain number of terms of Steklov-eigenfunctions. From left-to-right, the number of terms increases. Each plot will contain the partial sum of the previous result plus the next Steklov-eigenfunction multiplied against the coefficient $\langle s_k , g\rangle_\Sigma/\delta_k$. The 2nd and 3rd eigenfunction do not play a role in the reconstruction for this example. Approximately 3 Eigenfunctions are reproducing the qualitative features of the solution. The solution obtained by mixed FEM is shown in the bottom right hand corner for comparison.



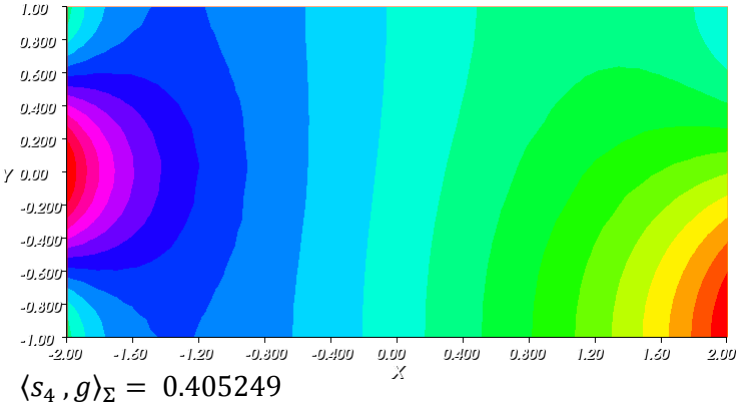
$\langle s_1 , g \rangle_\Sigma$ = -0.666667



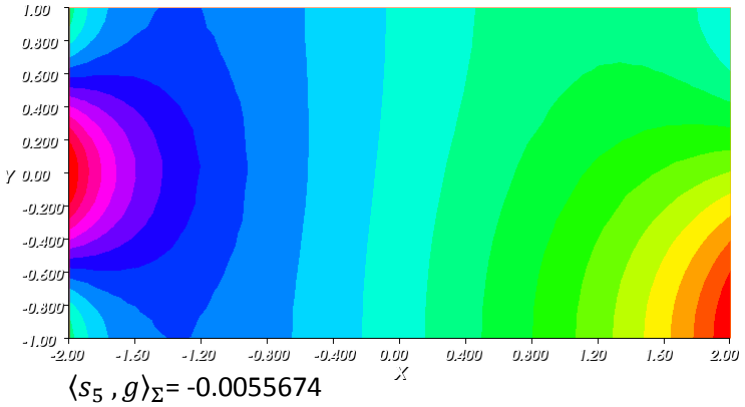
$\langle s_2 , g \rangle_\Sigma$= -0.245764



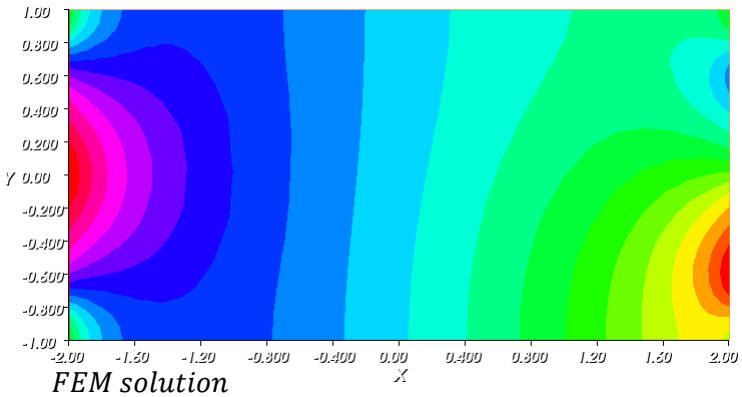
$\langle s_3 , g \rangle_\Sigma$ = -0.249078



$\langle s_4 , g \rangle_\Sigma = 0.405249$



$\langle s_5 , g \rangle_\Sigma$= -0.0055674



FEM solution

Figure 8: The plots show the reconstruction of the solution to the mixed DN bvp. Each plot contains a partial sum of a certain number of terms of Steklov-eigenfunctions. From left-to-right, the number of terms increases. Each plot will contain the predecessor result plus the next Steklov-eigenfunction multiplied against the coefficient $\langle s_k , g \rangle_\Sigma$. Most of the qualitative features are resolved by term 5. The solution using FEM is included in the bottom right hand corner for reference.

(vii) Measuring Convergence and Comparing SEM to FEM

It is important to measure how well an approximate solution u_n to a variational problem compares to its actual solution u^*. It is equally important to measure how well different approximations compare to each other. Comparison is usually performed by using a norm, generally related to the solution function space. The most common norms used for comparison are the L2-norm (10), H1-norm (11), and L_∞-norm $\|u\|_\infty = sup_{x\in\Omega} u(x)$. Each of these norms can be approximated by discrete versions in FreeFEM++. The L2-norm and H1-norm involve evaluating an integral of functions defined over the mesh T_h. Integrals can be calculated using the **int2d**, **int1d**, or **int3d** functions in FreeFEM++, depending on the dimension of the mesh. The L_∞-norm can be calculated by built in FreeFEM++ tools.

There are two types of error estimates in FEM, priori and posterior. Priori error estimates are useful for determining if a numerical algorithm will theoretically converge. Posterior estimates can be used to determine the current error residual in an approximate solution. This type of error estimate is more useful for computation as it can be used to adapt regions in the mesh that require refinement while keeping other regions coarse. FreeFEM++ can adapt a mesh through a function called **adaptmesh**. There is a general error indicator built in that utilizes the Hessian matrix. Custom error indicators can also be developed and passed to the adaptmesh routine. The mesh can also be manually refined by subdividing triangles into sub-triangles using the **splitmesh** routine. For the computations carried out in subsequent sections, combinations of the built-in routines are used to adapt the mesh to yield converged solutions. Convergence will be measured by carefully comparing successive approximations. This loosely works provided the mesh step-size is steadily decreasing enough to expect a large enough difference in the answer. If no change is detected within some norm-tolerance then the solution is considered converged. The following code how to calculate norms.

```
// Declare two functions to demonstrate norms
func g1 = (x/L)^2+(y/M)^2;
func g2 = ((L+x)*(y/M)*(1.0-y/M)+(x-L)*(1.0-(y/M)^2))/(2.0*L);

// Declare two functions to demonstrate norms, mesh Th assumed created
fespace Vh (Th,P1);
Vh gh1 = g1, gh2=g2;
Vh gDiff = gh1-gh2;

// Absolute differences
cout << "L2 Error : " <<  sqrt(int2d(Th)( gDiff ^2)) << end;
cout << "H1 Error : " <<  sqrt(int2d(Th)( gDiff ^2 + dx(gDiff)^2+dy(gDiff)^2));<< end;
cout << "L-Inf Error : " <<  gDiff[].max << end;
// Relative differences
cout << "L2 Rel g1 Error : " <<  sqrt(int2d(Th)( gDiff ^2))/ sqrt(int2d(Th)( gh1 ^2))  << end;
cout << "H1 Rel g1 Error : " <<  sqrt(int2d(Th)( gDiff ^2 + dx(gDiff)^2+dy(gDiff)^2))/
                                 sqrt(int2d(Th)( gh1 ^2 + dx(gh1)^2+dy(gh1)^2));<< end;
```

Code Snippet 8: Demonstrates how to calculate norms.

(viii) Mesh Adaptation

Higher-order Steklov-eigenfunctions tend to be localized along the boundaries of the domain. A smaller mesh-size must be used to capture this localization. A uniformly triangulated mesh becomes inefficient as localization causes sharp gradients in the function and the solutions on the interior of region are near constant. The localization can efficiently be captured by adapting the mesh properly to the Steklov-eigenfunctions. The mesh has to be adapted in such a way that it can capture the lower and higher order Steklov-eigenfunctions. The following code starts with a coarse mesh (coarse enough to resolve enough eigenfunctions), adapt the mesh against a function that combines information about the Steklov-eigenfunctions and continuously sub-divide triangles in half until convergence is achieved.

```
Vh[int] eVold(eigCount); // Last eigenvectors
real L2Error = 1.0E8;       // Tracks the total L2 error
real L2ETol  = 1.0E-3;      // Termination criteria
real errAdapt = 1.0E-3;    // Adaptation routine

int count = 0, maxTries = 10, numEigs = eigCount;
while(L2Error > L2ETol  && count < maxTries)
{
     if (count > 0 && count < 4) // Adapt mesh a few times
     {
          Vh fAdapt = 0;
          for(int i=1;i<numEigs;i++) // Construct a function that captures localization of all terms
               fAdapt = fAdapt +  max(fAdapt,abs(ev[i]^4*eV[i]));
          Th = adaptmesh(Th,fAdapt, errAdapt );
          errAdapt /=2.0;
     }  else if (count > 1)  Th = splitmesh(Th,2); // Subdivide mesh in half

     // Get resulting matrices Ax=lBx
     matrix A = va(Vh,Vh,solver = sparsesolver);
     matrix B = vb(Vh,Vh);
     // Solve Ax=lBx
     numEigs = EigenValue(A,B,sym=true,sigma=0,value=ev,vector=eV);
     numEigs = min(eigCount,numEigs);

     L2Error = 0.0;
     for(int i=1;i<numEigs;i++) // Capture worst L2 error between eigenfunctions and last set
     {
          Vh ssDiff = abs(abs(eV[i])-abs(eVold[i]));
          L2Error = max(L2Error, int2d(Th)(ssDiff*ssDiff));
          eVold[i] = eV[i];
     }
     count++;
}
```

Code Snippet 9: Demonstrates how to use adapt the mesh to efficiently capture low-order and high-order Steklov-eigenfunctions.

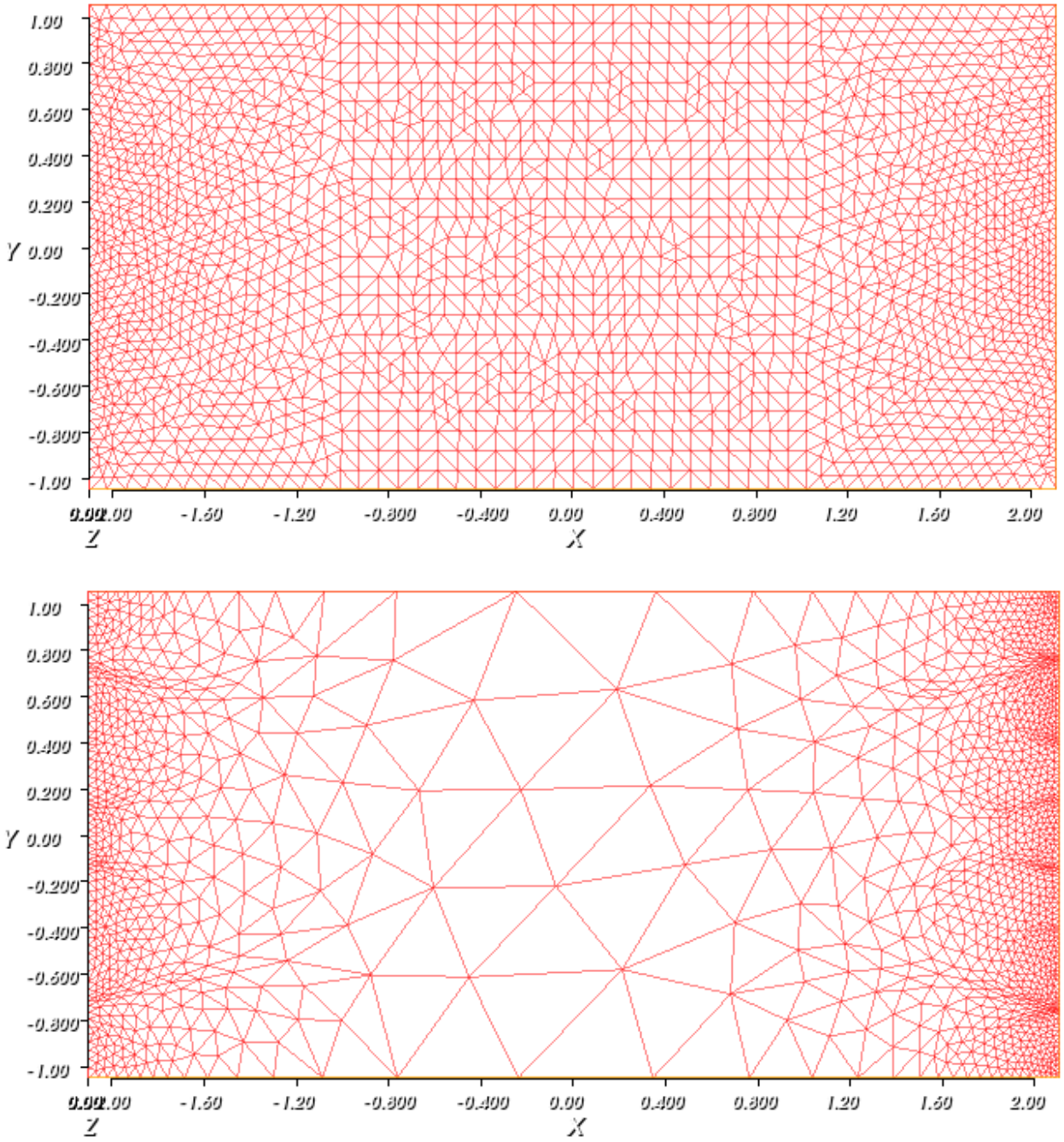


Figure 9: Top – original uniform mesh. Bottom - Adapted and refined mesh. Notice the finer mesh near the boundaries capture the localization.

8. Comparison of FEM vs. SEM for Five Model Problems

Armed with Section 6, five model problems are solved using SEM and FEM (or mixed FEM). Results are plotted and compared between methods.

(i) Heat Conduction in a Solid Plate

A classical mixed DN BVP is steady heat conduction in a $2Lx2M$ solid plate $(L >= M)$ made of an isotropic material. The left and right edges of the plate have prescribed temperature while the bottom and top edges are perfectly insulated; the gradient of the temperature is zero. The problem takes the classical form:

Let $u(x)$ be the unknown temperature of the plate $\Omega = (-\mathrm{L}, \mathrm{L}) \times (-\mathrm{M}, \mathrm{M})$. $u(x)$ satisfies

$$\Delta u(x) = 0, \;\; x \in \Omega \tag{52}$$

subject to the mixed Dirichlet and Neumann boundary conditions

$$u(-L, y) = g_1(y) = (2 + cos\left(\frac{\pi y}{12M}\right)) * (1 - (\frac{y}{M})^2)\,, \tag{53}$$

$$u(L, y) = g_2(y) = |M - y|, for \;\; y \in [-M, M],$$

$$\frac{\partial u}{\partial y}(x, M) = \frac{\partial u}{\partial y}(x, -M) = 0, \;\; for\; x \in [-L, L]. \tag{54}$$

The equivalent variational problem is given by

Find $u \in H^1_{\Sigma g}(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla \mathrm{v}\, dx = 0, \text{ for all } v \in H^1_{\Sigma 0}(\Omega)\,,$$

where $H^1_{\Sigma g}(\Omega) \coloneqq \{u \in H^1(\Omega) \colon u(-L, y) = g_1(y) = and\; u(L, y) = g_2(y), for \;\; y \in [-M, M]\}$.

A problem of this type has already been solved in section 6. Here the boundary conditions have been changed. The next code sample demonstrates the methods discussed in section 6 and solves problem (52-54) using both FEM and SEM. Comparison results will also be shown for different geometry sizes. The Steklov-eigenvalue and eigenfunction spectrums of the SEM method are also shown and compared against analytical formula derived in section 5.

```
macro Grad(u) [dx(u),dy(u)]
// Geometry - Solid Plate
real L = 2.0, M=1.0;  // L - value
border Left (t=0,1) {x=-L;y=M*(1.0-2.0*t);label=1;}  // Left barrier
border Top  (t=0,1) {x=L*(1.0-2.0*t);y=M;label=2;}   // Top wall
border Right(t=0,1) {x=L;y=M*(2.0*t-1.0);label=3;}   // Right barrier
border Bottom(t=0,1){x=L*(2.0*t-1.0);y=-M;label=2;}  // Bottom wall

// Construct a mesh
mesh Th = buildmesh(Left(49) + Top(49) + Right(49) + Bottom(49));

// Define inlet/outlet
func g1 = (2.0+cos(2.0*pi*y/M))*(1.0-(y/M)^2);
func g2 = M-abs(y);

// Finite element and functions
fespace Vh(Th,P2);
Vh uhFEM,uhSEM,vh;                          // Holds the final uhFEM
int eigCount=54, numEigs=54, count=0; // total number of eigenvalues
real[int] ev(eigCount);                     // Eigenvalues
Vh [int]   eV(eigCount),eVold(eigCount),pSum(eigCount);  // Eigenvectors

// Steklov - Eigenvalue problem in variational form
real L2Errorsq = 1.0E8, adaptErr = 1.0e-1, shift = 10;
varf vA(u,v) = int1d(Th,1,3)( -shift*u*v) +int2d(Th)(dx(u)*dx(v)+dy(u)*dy(v));
varf vB(u,v) = int1d(Th,1,3)( u*v);
while(L2Errorsq > 2.0E-2)
{
  if (count > 0 && count < 3) // Adaptation step
  {
     Vh fAdapt = 0;
     for(int i=0;i<numEigs;i++)
        fAdapt = fAdapt +  abs(eV[i]);
     Th = adaptmesh(Th,fAdapt,err=adaptErr);
     adaptErr/=2.0;
  } else if (count > 0) Th = splitmesh(Th,2);

  // Get resulting matrices Ax=lBx
  matrix A = vA(Vh,Vh,solver = sparsesolver);
  matrix B = vB(Vh,Vh);
```

```
  // Solve Ax=lBx
  numEigs = EigenValue(A,B,sym=true,sigma=shift,value=ev,vector=eV);
  numEigs = min(eigCount,numEigs);
  L2Errorsq = 0.0;
  for(int i=0;i<numEigs;i++)
  {
     Vh ssDiff = abs(abs(eV[i])-abs(eVold[i]));
     L2Errorsq = max(L2Errorsq,int2d(Th)(ssDiff*ssDiff));
     eVold[i] = eV[i];
  }
  count++;
}
// Construct solution
real[int] c(numEigs);
for(int i=0;i<numEigs;i++)
{
  c[i] =int1d(Th,1)(g1*eV[i]) + int1d(Th,3)(g2*eV[i]);
  uhSEM = uhSEM + c[i] * eV[i];
  pSum[i] = uhSEM;
}

// Solve mixed DN FEM problem
solve mixedDNFEM(uhFEM,vh)=int2d(Th)(Grad(uhFEM)'*Grad(vh)) +
                                   on(1,uhFEM=g1) +  on(3,uhFEM=g2);
// Plot solutions
plot(uhSEM,value=true,fill=true,ColorScheme=2);
plot(uhFEM,value=true,fill=true,ColorScheme=2);
```

Code Snippet 10: Code to solve mixed DN BVP with FEM and adapted SEM.

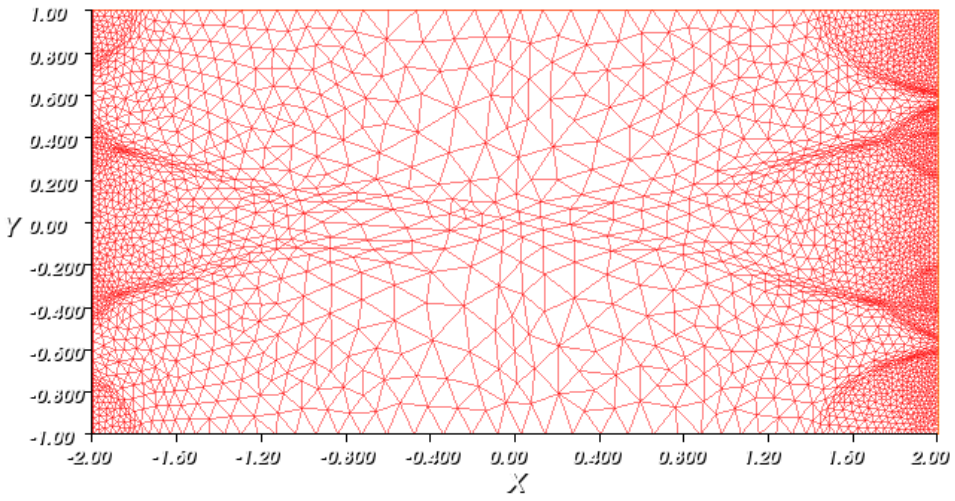


Figure 10: Adapted mesh for SEM method

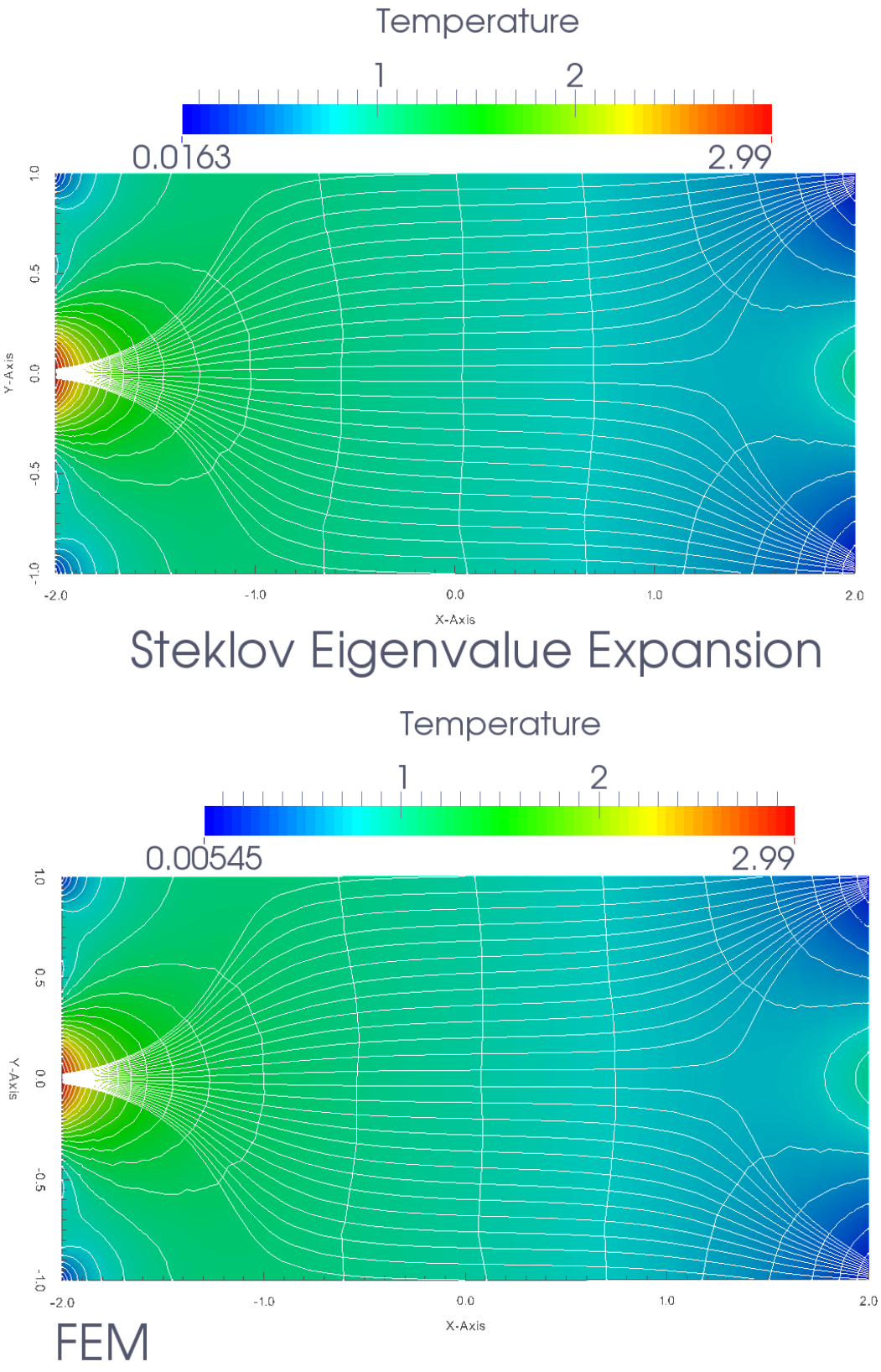


Figure 11: Solutions from both SEM & FEM procedures. Horizontal particle streamlines and vertical potential lines are overlayed the solution.

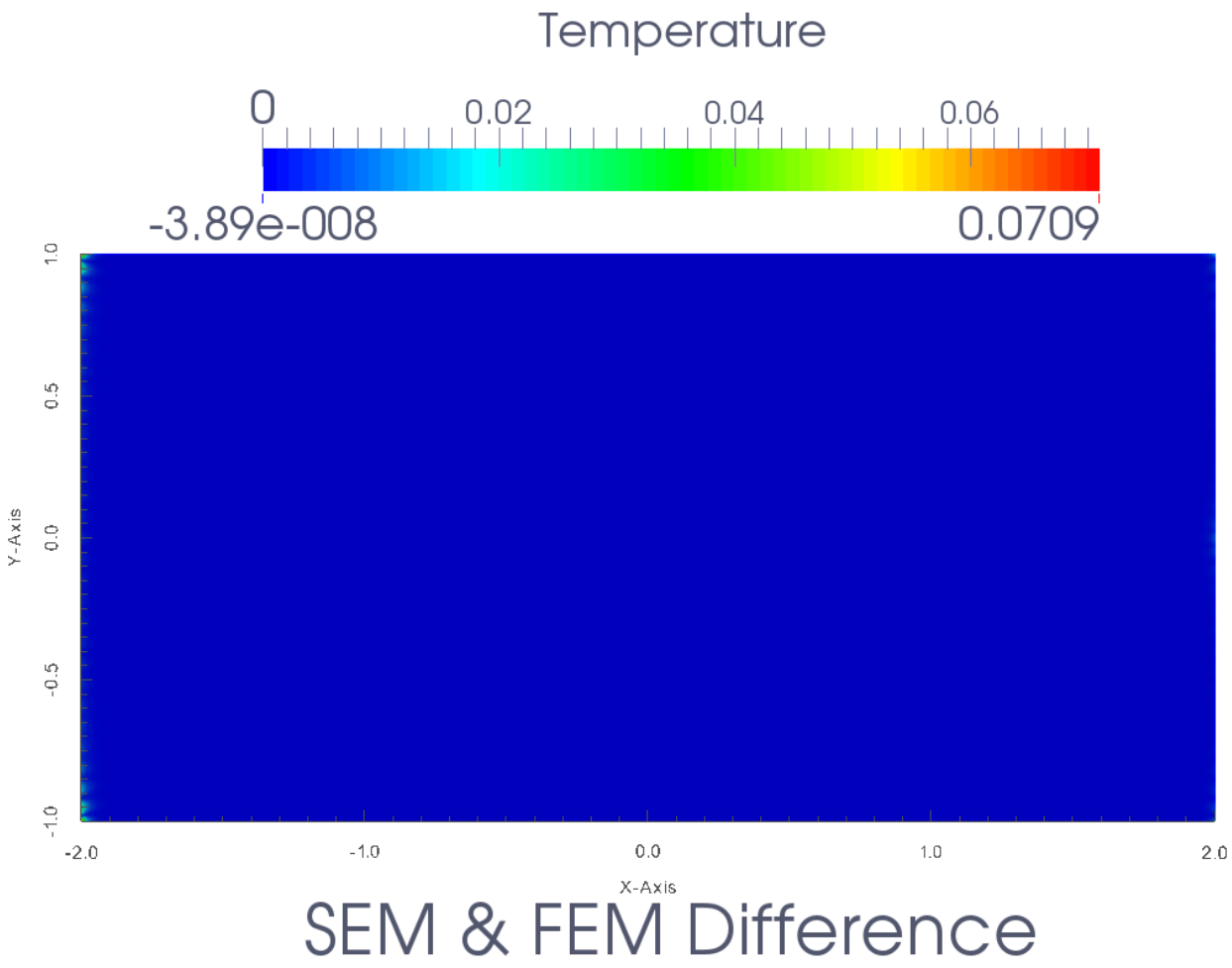


Figure 12: Absolute difference of SEM & FEM solutions. Some difference is revealed on the boundary. Additional refinement might be required on boundary or more modes need to be captured.

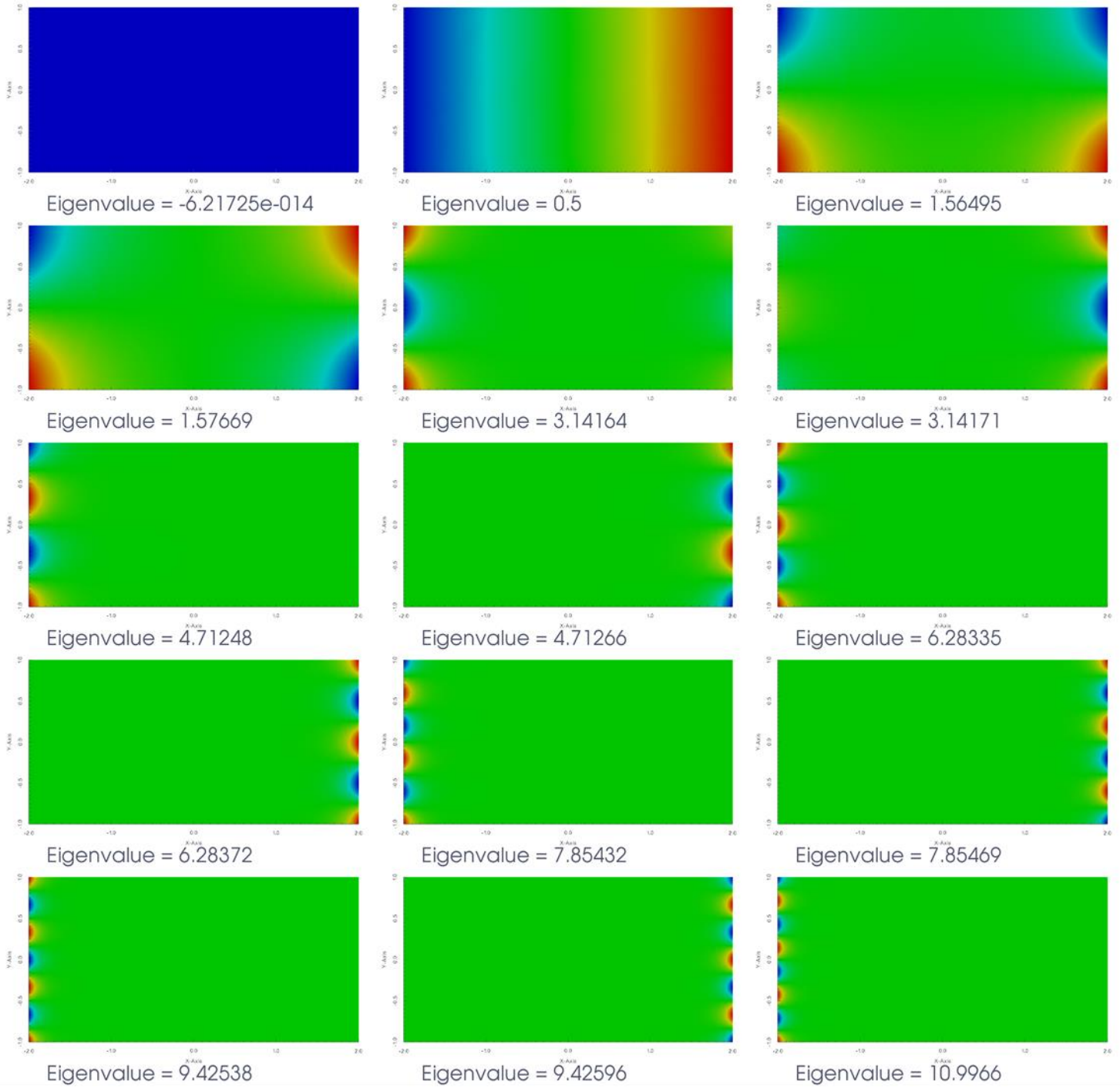


Figure 13: First 15 ordered Steklov-eigenfunctions (left to right) of the Steady Heat Conduction on a 4x2 Solid Plate problem.

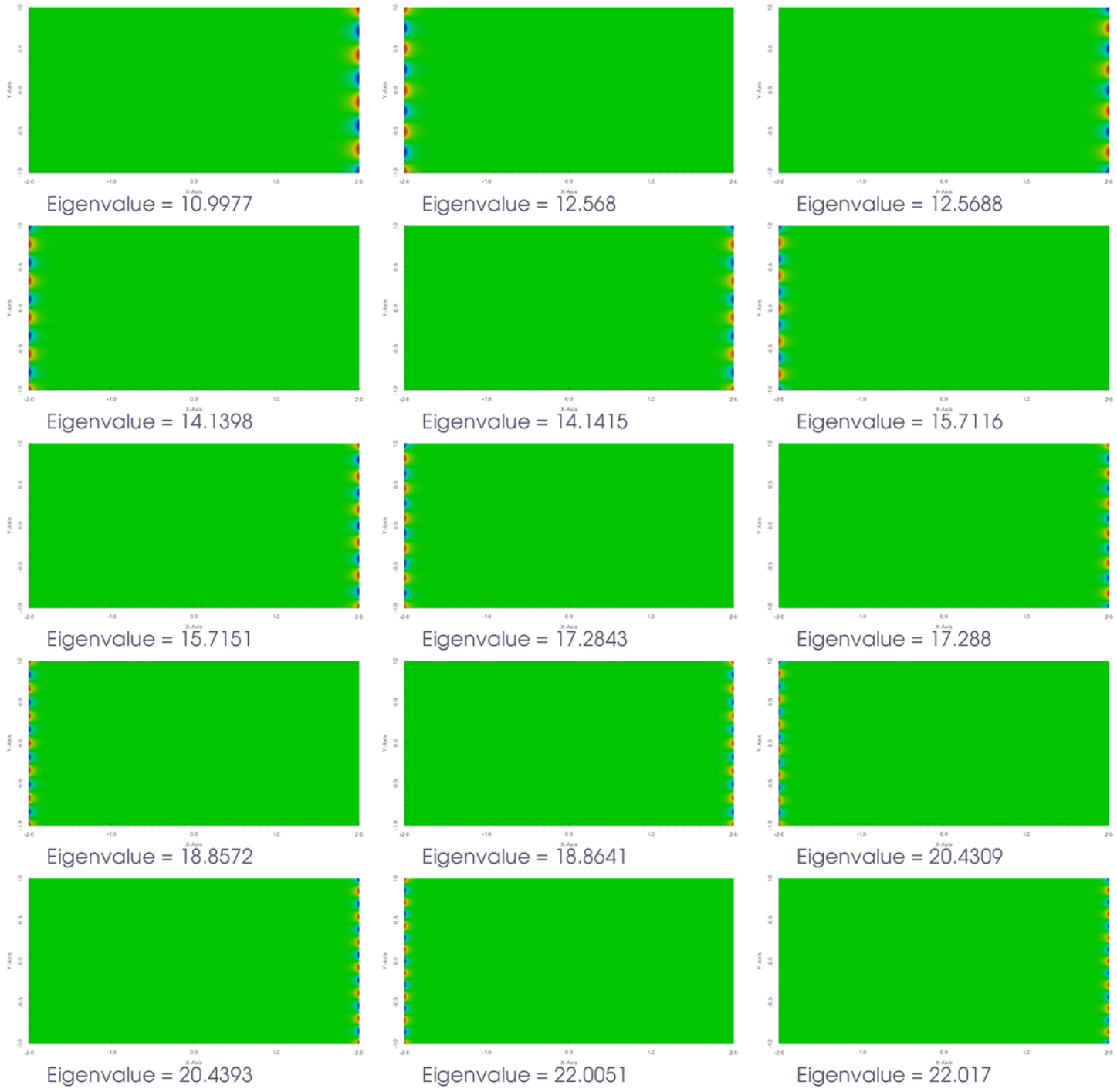


Figure 14: Next 15 ordered Steklov eigenfunctions (left to right) of the Steady Heat Conduction on a 4x2 Solid Plate problem. Notice the higher order Steklov-eigenfunctions are localized near the boundary.

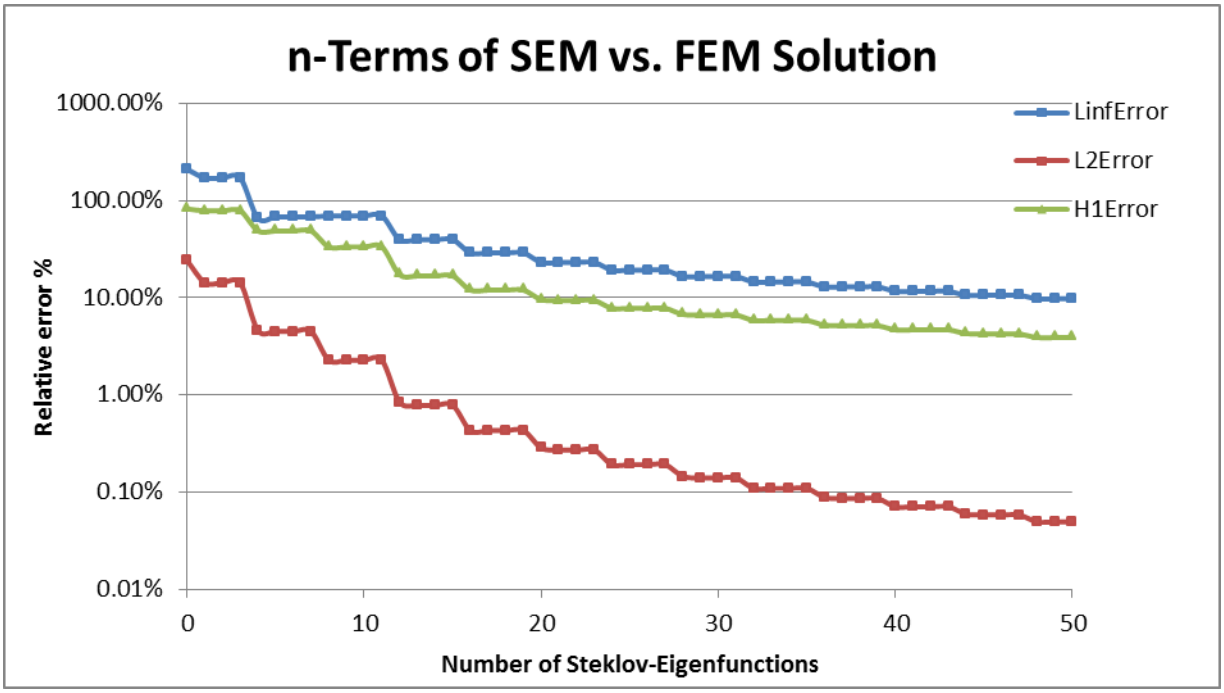


Figure 15: Relative errors for different norms comparing n-successive terms in the Steklov-expansion (x-axis) compared against solution produced by FEM (y-axis).

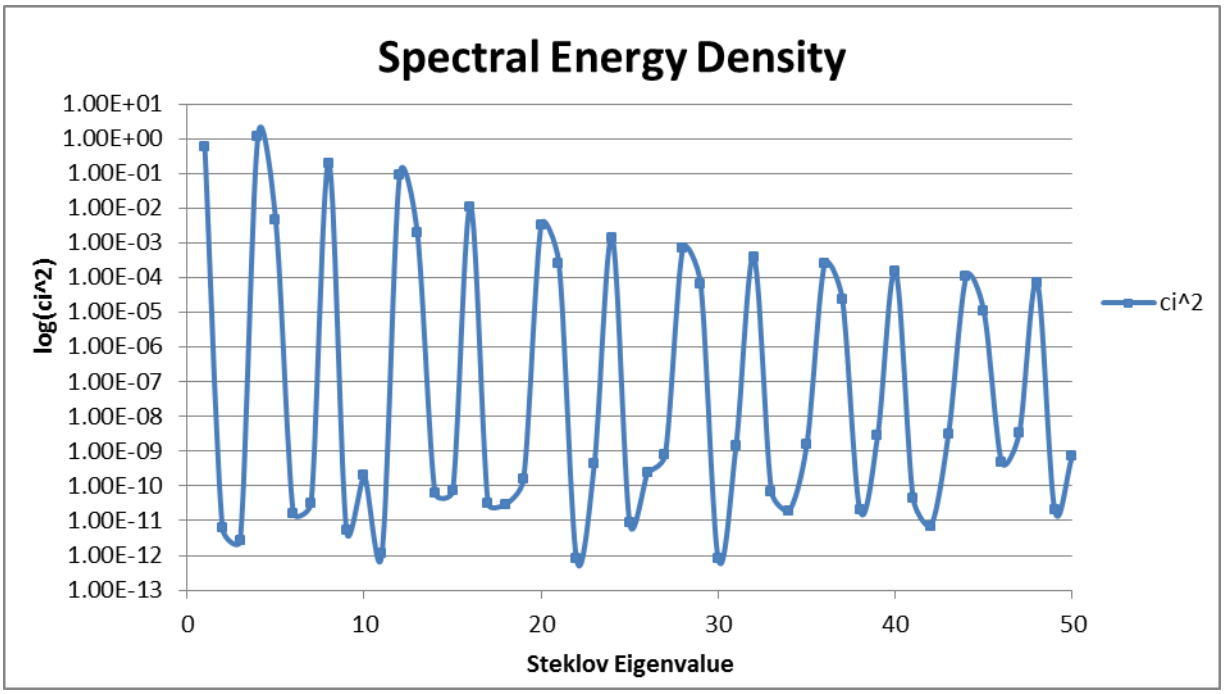


Figure 16: Spectral energy density given by $\langle s_n, g \rangle_{\Sigma}^2$.

	Steklov Eigenvalue	Relative L_∞-Error n SEM Terms Vs FEM Solution	Relative $L_2 - Error$ n SEM Terms Vs FEM Solution	Relative $H_1 - Error$ n SEM Terms Vs FEM Solution	n Coefficient in SEM Expansions
n	δ_n	$\frac{\|u_n^s - u_{FEM}^*\|_\infty}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^s - u_{FEM}^*\|_2}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^s - u_{FEM}^*\|_{1,2}}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\langle s_n, g\rangle_\Sigma$
0	-6.22E-14	2.10865	0.242433	0.825129	-1.78267
1	0.5	1.71731	0.143154	0.787987	-0.78268
2	1.56495	1.71731	0.143154	0.787987	2.45E-06
3	1.57669	1.71731	0.143154	0.787987	-1.64E-06
4	3.14164	0.660514	0.045947	0.489495	-1.11006
5	3.14171	0.681585	0.04517	0.488529	-0.06885
6	4.71248	0.681585	0.04517	0.488529	-4.07E-06
7	4.71266	0.681585	0.04517	0.488529	5.67E-06
8	6.28335	0.698315	0.022828	0.33467	0.451355
9	6.28372	0.698315	0.022828	0.33467	-2.30E-06
10	7.85432	0.698316	0.022828	0.334669	-1.40E-05
11	7.85469	0.698316	0.022828	0.334669	-1.06E-06
12	9.42538	0.397469	0.008441	0.173486	-0.30082
13	9.42596	0.397469	0.007821	0.168101	0.045029
14	10.9966	0.397461	0.007821	0.168102	-7.94E-06
15	10.9977	0.397461	0.007821	0.168102	-8.49E-06
16	12.568	0.290466	0.004302	0.120211	-0.10697
17	12.5688	0.290466	0.004302	0.120212	5.60E-06
18	14.1398	0.290472	0.004302	0.120211	-5.47E-06
19	14.1415	0.290472	0.004302	0.120211	-1.27E-05
20	15.7116	0.231391	0.002848	0.095981	0.059053
21	15.7151	0.231391	0.002707	0.093812	-0.0162
22	17.2843	0.23139	0.002707	0.093812	8.85E-07
23	17.288	0.23139	0.002707	0.093812	-2.11E-05
24	18.8572	0.193032	0.001917	0.078082	-0.03834
25	18.8641	0.193032	0.001917	0.078082	-2.90E-06
26	20.4309	0.193028	0.001917	0.078081	1.53E-05
27	20.4393	0.193028	0.001917	0.078082	-2.86E-05
28	22.0051	0.165828	0.001451	0.06754	-0.02715
29	22.017	0.165828	0.0014	0.066451	0.00827

Table 2: Steklov Eigenvalues for Steady Heat Conduction problem on a (2x1) solid plate. Relative errors in different norms captured for n-successive terms in the Steklov-expansion compared against solution produced by FEM.

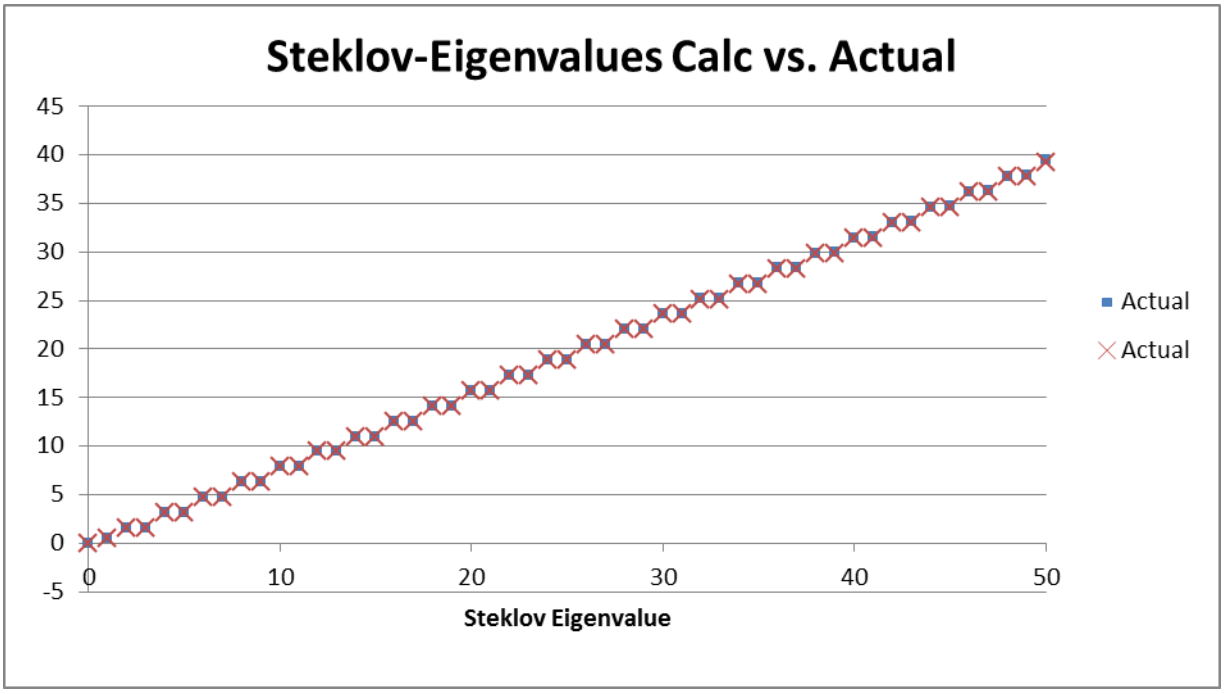


Figure 17: Comparison plot of calculated eigenvalues by SEM versus actual formula for eigenvalues for rectangle case.

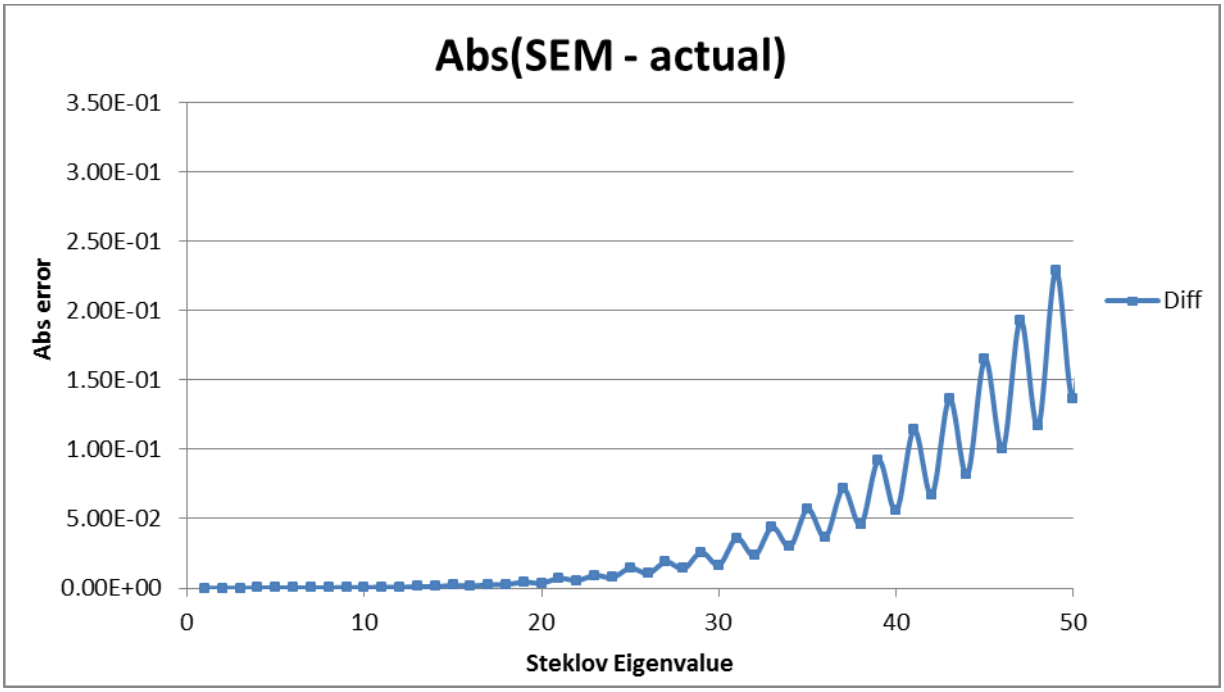


Figure 18: Absolute difference between calculated eigenvalues by SEM and actual eigenvalues.

	Steklov Eigenvalue SEM	Analytical Solution	Absolute Difference
n	δ_n	$\hat{\delta}_n$	$\lvert\delta_n - \hat{\delta}_n\rvert$
0	-6.22E-14	0.0	6.21725E-14
1	0.5	0.5	0.00000E+00
2	1.56495	1.56494	9.48218E-06
3	1.57669	1.57667	1.59525E-05
4	3.14164	3.14157	6.92580E-05
5	3.14171	3.14161	9.54347E-05
6	4.71248	4.71239	9.10810E-05
7	4.71266	4.71239	2.70958E-04
8	6.28335	6.28319	1.64693E-04
9	6.28372	6.28319	5.34693E-04
10	7.85432	7.85398	3.38366E-04
11	7.85469	7.85398	7.08366E-04
12	9.42538	9.42478	6.02039E-04
13	9.42596	9.42478	1.18204E-03
14	10.9966	10.99557	1.02571E-03
15	10.9977	10.99557	2.12571E-03
16	12.568	12.56637	1.62939E-03
17	12.5688	12.56637	2.42939E-03
18	14.1398	14.13717	2.63306E-03
19	14.1415	14.13717	4.33306E-03
20	15.7116	15.70796	3.63673E-03
21	15.7151	15.70796	7.13673E-03
22	17.2843	17.27876	5.54041E-03
23	17.288	17.27876	9.24041E-03
24	18.8572	18.84956	7.64408E-03
25	18.8641	18.84956	1.45441E-02
26	20.4309	20.42035	1.05478E-02
27	20.4393	20.42035	1.89478E-02
28	22.0051	21.99115	1.39514E-02
29	22.017	21.99115	2.58514E-02

Table 3: Table of Steklov eigenvalues calculated by SEM method and actual Steklov-eigenvalues by analytical formula for the Steady Heat Conduction problem on a (2x1) solid plate. Notice the error increases for larger terms due to the difficulty of calculating the Steklov-eigenfunctions localized to the boundary.

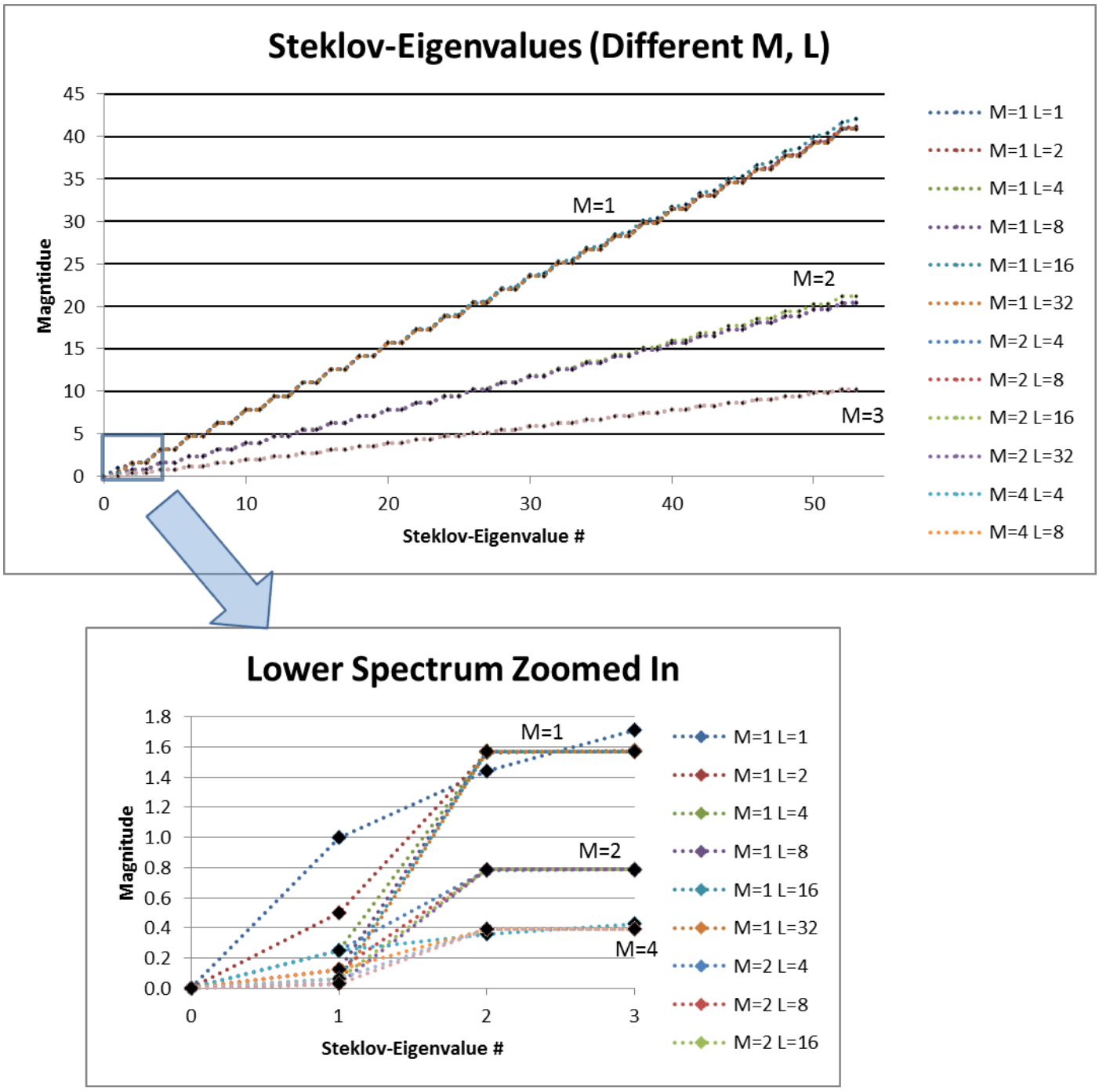


Figure 19: Steklov-Eigenvalue spectra for different rectangle sizes L>=M. M appears to be the dominating parameter.

M	1	1	1	1	1	1	2	2	2	2	4	4	4	4
L	1	2	4	8	16	32	4	8	16	32	4	8	16	32
#														
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	1.00	0.50	0.25	0.13	0.06	0.03	0.25	0.13	0.06	0.03	0.25	0.13	0.06	0.03
2	1.44	1.56	1.57	1.57	1.57	1.57	0.78	0.79	0.79	0.79	0.36	0.39	0.39	0.39
3	1.71	1.58	1.57	1.57	1.57	1.57	0.79	0.79	0.79	0.79	0.43	0.39	0.39	0.39
4	3.13	3.14	3.14	3.14	3.14	3.14	1.57	1.57	1.57	1.57	0.78	0.79	0.79	0.79
5	3.15	3.14	3.14	3.14	3.14	3.14	1.57	1.57	1.57	1.57	0.79	0.79	0.79	0.79
6	4.71	4.71	4.71	4.71	4.71	4.71	2.36	2.36	2.36	2.36	1.18	1.18	1.18	1.18
7	4.71	4.71	4.71	4.71	4.71	4.71	2.36	2.36	2.36	2.36	1.18	1.18	1.18	1.18
8	6.28	6.28	6.28	6.28	6.29	6.28	3.14	3.14	3.14	3.14	1.57	1.57	1.57	1.57
9	6.28	6.28	6.28	6.28	6.29	6.28	3.14	3.14	3.14	3.14	1.57	1.57	1.57	1.57
10	7.85	7.85	7.85	7.85	7.86	7.85	3.93	3.93	3.93	3.93	1.96	1.96	1.96	1.96
11	7.85	7.85	7.85	7.85	7.87	7.85	3.93	3.93	3.93	3.93	1.96	1.96	1.96	1.96
12	9.42	9.43	9.42	9.43	9.44	9.42	4.71	4.71	4.72	4.71	2.36	2.36	2.36	2.36
13	9.42	9.43	9.42	9.43	9.44	9.42	4.71	4.71	4.72	4.71	2.36	2.36	2.36	2.36
14	11.00	11.00	11.00	11.00	11.01	11.00	5.50	5.50	5.50	5.50	2.75	2.75	2.75	2.75
15	11.00	11.00	11.00	11.00	11.03	11.00	5.50	5.50	5.51	5.50	2.75	2.75	2.75	2.75
16	12.57	12.57	12.57	12.57	12.59	12.57	6.28	6.28	6.29	6.28	3.14	3.14	3.14	3.14
17	12.57	12.57	12.57	12.57	12.61	12.57	6.28	6.28	6.30	6.28	3.14	3.14	3.14	3.14
18	14.14	14.14	14.14	14.14	14.17	14.14	7.07	7.07	7.08	7.07	3.53	3.53	3.53	3.53
19	14.14	14.14	14.14	14.14	14.21	14.14	7.07	7.07	7.09	7.07	3.53	3.53	3.53	3.53
20	15.71	15.71	15.71	15.71	15.76	15.71	7.85	7.85	7.87	7.85	3.93	3.93	3.93	3.93
21	15.71	15.72	15.71	15.71	15.80	15.71	7.85	7.85	7.88	7.85	3.93	3.93	3.93	3.93
22	17.28	17.28	17.28	17.28	17.34	17.28	8.64	8.64	8.67	8.64	4.32	4.32	4.32	4.32
23	17.28	17.29	17.28	17.28	17.39	17.28	8.64	8.64	8.67	8.64	4.32	4.32	4.32	4.32
24	18.85	18.86	18.85	18.85	18.93	18.85	9.42	9.42	9.46	9.43	4.71	4.71	4.71	4.71
25	18.85	18.86	18.85	18.85	19.01	18.85	9.42	9.43	9.48	9.43	4.71	4.71	4.71	4.71
26	20.42	20.43	20.42	20.43	20.51	20.42	10.21	10.21	10.26	10.21	5.11	5.11	5.11	5.11
27	20.42	20.44	20.42	20.43	20.61	20.42	10.21	10.21	10.28	10.21	5.11	5.11	5.11	5.11
28	21.99	22.01	21.99	22.00	22.11	21.99	11.00	11.00	11.07	11.00	5.50	5.50	5.50	5.50
29	21.99	22.02	21.99	22.00	22.22	21.99	11.00	11.00	11.08	11.00	5.50	5.50	5.50	5.50

Table 4: Summary of Steklov-eigenvalue spectra for different rectangle sizes L>=M.

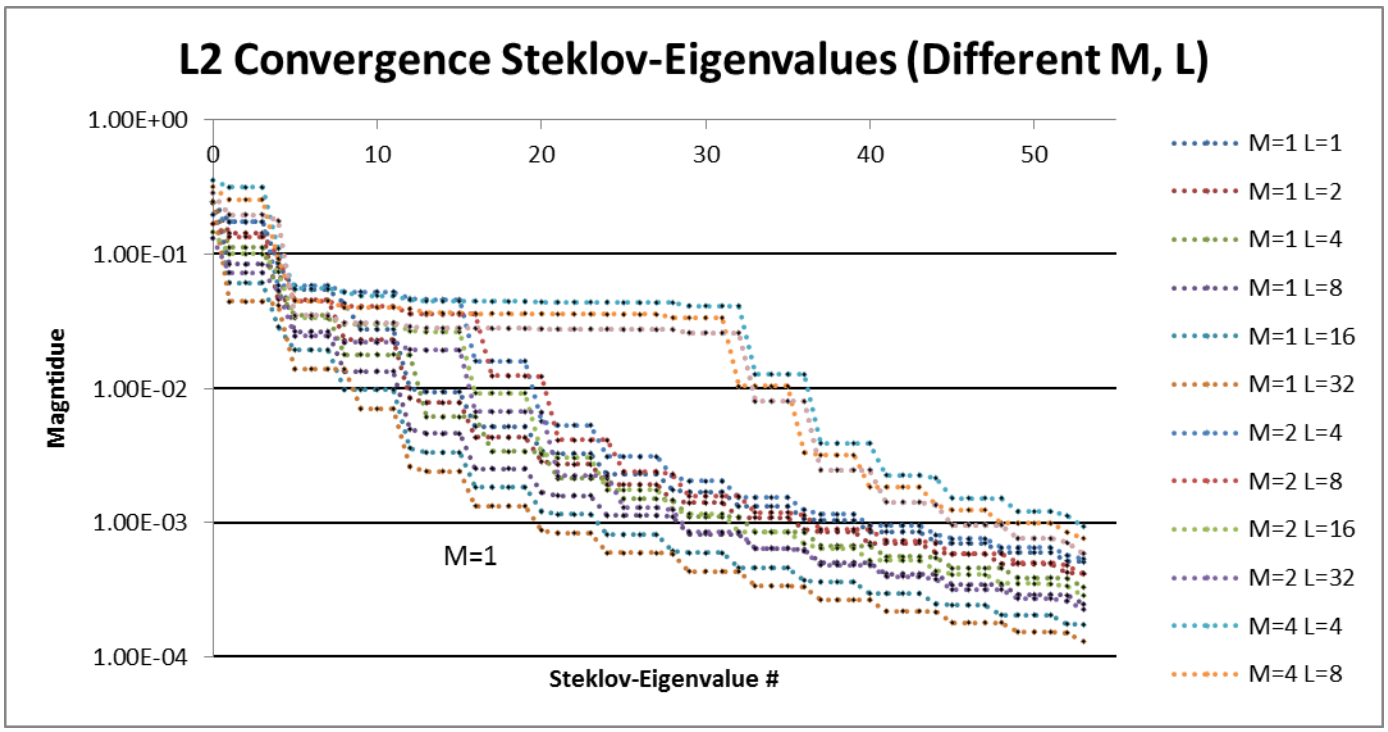


Figure 20: Different L2-Norm convergence rates for different geometries. Notice the M parameter has an effect on convergence.

(ii) Electrostatics

The FreeFEM++ manual provides an example of a Dirichlet BVP in electrostatics using FEM; see section 9.1.2 in [1]. This demonstration models the same problem with the exception of replacing a zero potential Dirichlet boundary condition on the outer wall with a no-flux boundary condition. The geometry consists of a circular conductor $\mathrm{C}_0 \coloneqq \{(5\cos t\,, 5\sin t)|\, t \in [0,1]\}$ containing two elliptical oppositely charged holes,

$\mathrm{C}_1 \coloneqq \{(2 + 0.3\cos t\,, 3\sin t)|\, t \in [0,1]\}$ and $\mathrm{C}_2 \coloneqq \{(-2 + 0.3\cos t\,, 3\sin t)|\, t \in [0,1]\}$.

The charge is represented by constant Dirichlet conditions on the boundary of the holes. It is assumed that there is no current and charge distribution is independent of time. The problem takes the form:

Let $\varphi(x)$ be the electric potential of the conductor, $\Omega = \mathrm{C}_0 \backslash (C_1 \cup C_2)$. $\varphi(x)$ satisfies

$$\Delta\varphi(x) = 0, \quad x \in \Omega \tag{55}$$

subject to the mixed Dirichlet and Neumann boundary conditions

$$\varphi(x,y) = g_1(x,y) = 1, for\ (x,y) \in C_1, \tag{56}$$

$$\varphi(x,y) = g_2(x,y) = -1, for\ (x,y) \in C_2,$$

$$\frac{\partial\varphi}{\partial v}(x,y) = 0\,, for\ (x,y) \in C_0. \tag{57}$$

The equivalent variational problem is given by

Find $u \in H^1_{\Sigma g}(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla \mathrm{v}\, dx = 0, \text{ for all } v \in H^1_{\Sigma 0}(\Omega)\,,$$

where $H^1_{\Sigma g}(\Omega) \coloneqq \{u \in H^1(\Omega) | \varphi(z) = g_1(z), for\ z \in C_1, \varphi(z) = g_2(z), for\ z \in C_2\}$.

```
macro Grad(u) [dx(u),dy(u)]
// Geometry - Solid Plate
real L = 2.0, M=1.0;  // L - value
border C0 (t=0,1) {x=5.0*cos(2.0*pi*t);y=5.0*sin(2.0*pi*t);}        // Circular conductor
border C1  (t=0,1) {x=2.0+0.3*cos(2.0*pi*t);y=3.0*sin(2.0*pi*t);} // Right conductor positive charge
border C2(t=0,1) {x=-2.0+0.3*cos(2.0*pi*t);y=3.0*sin(2.0*pi*t);}  // Left conductor negative charge

// Construct a mesh
mesh Th = buildmesh(C0(100) + C1(-200) + C2(-200));

// Define inlet/outlet
func g1 =  1;
func g2 = -1;

// Finite element and functions
fespace Vh(Th,P2);
Vh uhFEM,uhSEM,vh;                          // Holds the final uhFEM
int eigCount=75, numEigs=75, count=0; // total number of eigenvalues
real[int] ev(eigCount);                     // Eigenvalues
Vh [int]   eV(eigCount),eVold(eigCount),pSum(eigCount);  // Eigenvectors

// Steklov - Eigenvalue problem in variational form
real L2Errorsq = 1.0E8, adaptErr = 1.0e-2, shift = 0;
varf vA(u,v) = int1d(Th, C1,C2)(-shift* u*v)+ int2d(Th)(dx(u)*dx(v)+dy(u)*dy(v));
varf vB(u,v) = int1d(Th, C1,C2)( u*v);
while(L2Errorsq > 5.0E-1)
{
  if (count > 0 && count < 5) // Adaptation step
  {
     Vh fAdapt = 0;
     for(int i=0;i<numEigs;i++)
        fAdapt = fAdapt +  abs(eV[i]);
     Th = adaptmesh(Th,fAdapt,err=adaptErr);
     adaptErr/=2.0;
  } else if (count > 0) Th = splitmesh(Th,2);

  // Get resulting matrices Ax=lBx
  matrix A = vA(Vh,Vh,solver = sparsesolver);
  matrix B = vB(Vh,Vh);
```

```
  // Solve Ax=lBx
  numEigs = EigenValue(A,B,sym=true,sigma=0,value=ev,vector=eV,tol=1.0e-10);
  numEigs = min(eigCount,numEigs);
  L2Errorsq = 0.0;
  for(int i=0;i<numEigs;i++)
  {
     Vh ssDiff = abs(abs(eV[i])-abs(eVold[i]));
     L2Errorsq = max(L2Errorsq,int2d(Th)(ssDiff*ssDiff));
     eVold[i] = eV[i];
  }  count++;
}
// Construct solution
real[int] c(numEigs);
for(int i=0;i<numEigs;i++)
{
  c[i] =int1d(Th, C1)(g1*eV[i]) + int1d(Th, C2)(g2*eV[i]);
  uhSEM = uhSEM + c[i] * eV[i];
  pSum[i] = uhSEM;
}
// Solve mixed DN FEM problem
solve mixedDNFEM(uhFEM,vh)=int2d(Th)(Grad(uhFEM)'*Grad(vh)) +
                                   on(C1,uhFEM=g1) +  on(C2,uhFEM=g2);
// Plot solutions
plot(uhSEM,value=true,fill=true,ColorScheme=2);
plot(uhFEM,value=true,fill=true,ColorScheme=2);
```

Code Snippet 11: Code to solve mixed DN BVP with FEM and adapted SEM.

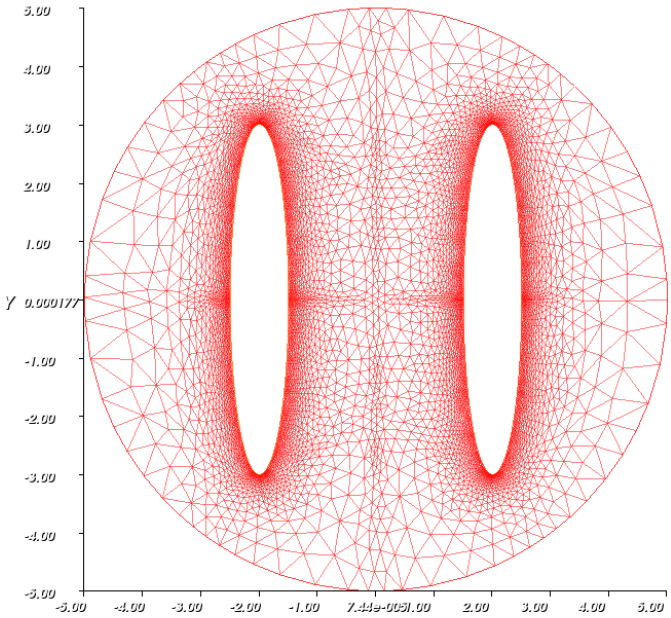


Figure 21: Adapted mesh constructed for the SEM method.

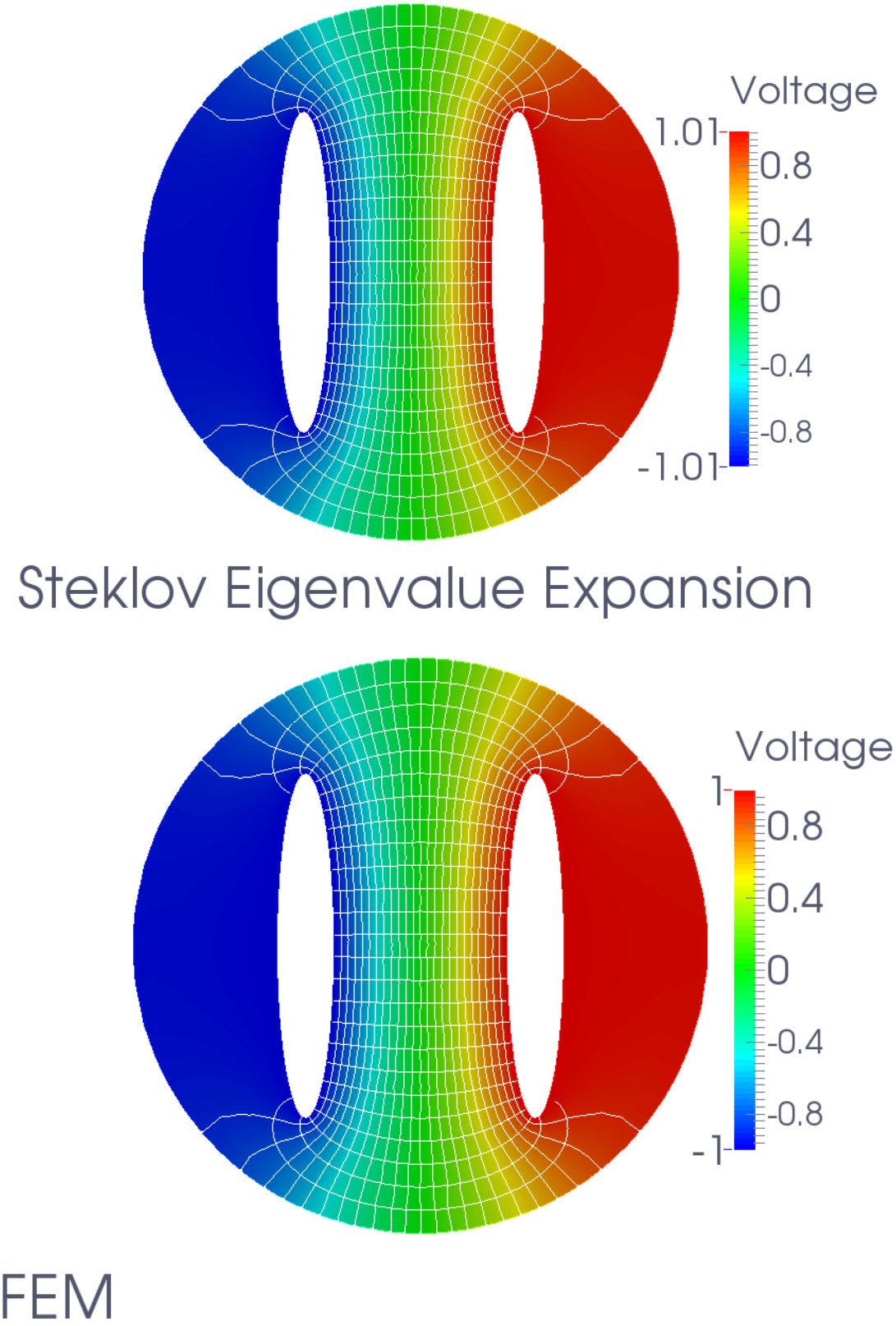


Figure 22: Solutions from both SEM & FEM procedures. Horizontal particle streamlines and vertical potential lines overlayed.

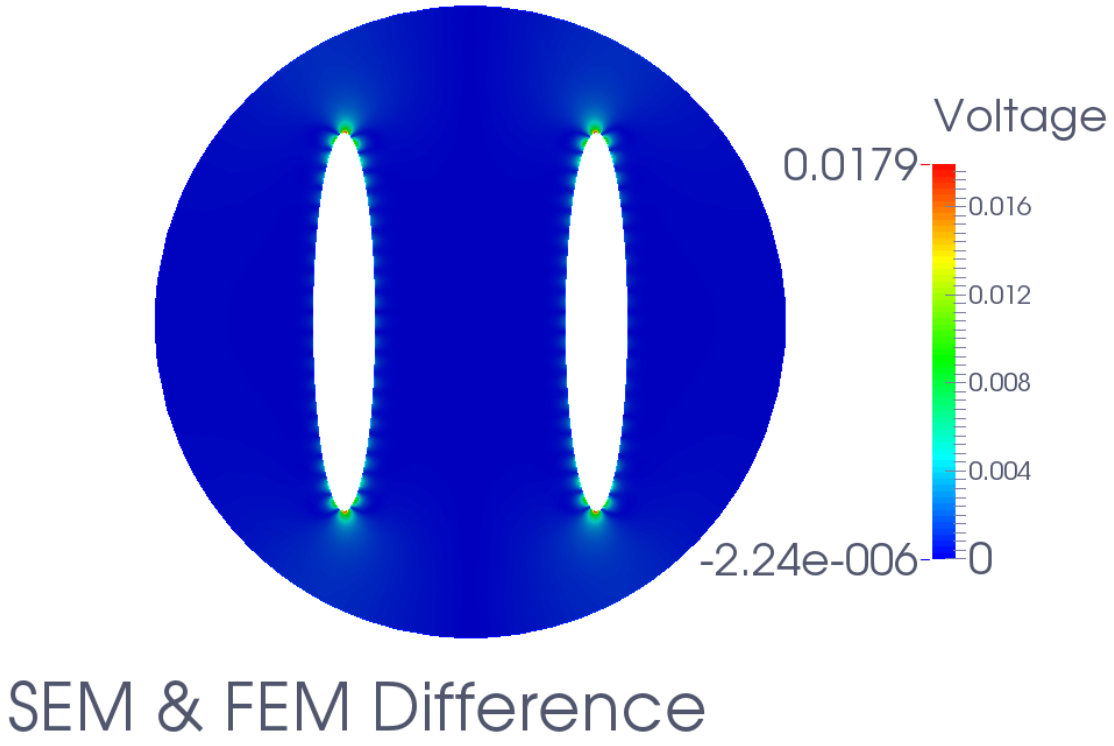


Figure 23: Absolute difference of SEM & FEM solutions. Some difference is revealed on the boundary.

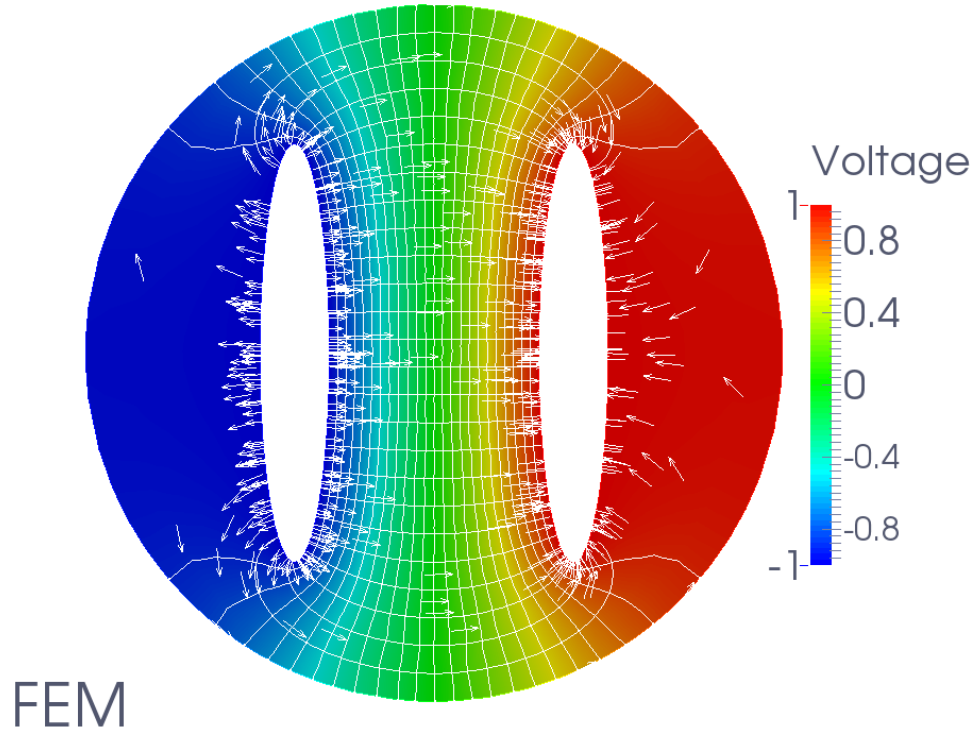


Figure 24: Electric field in conductor.

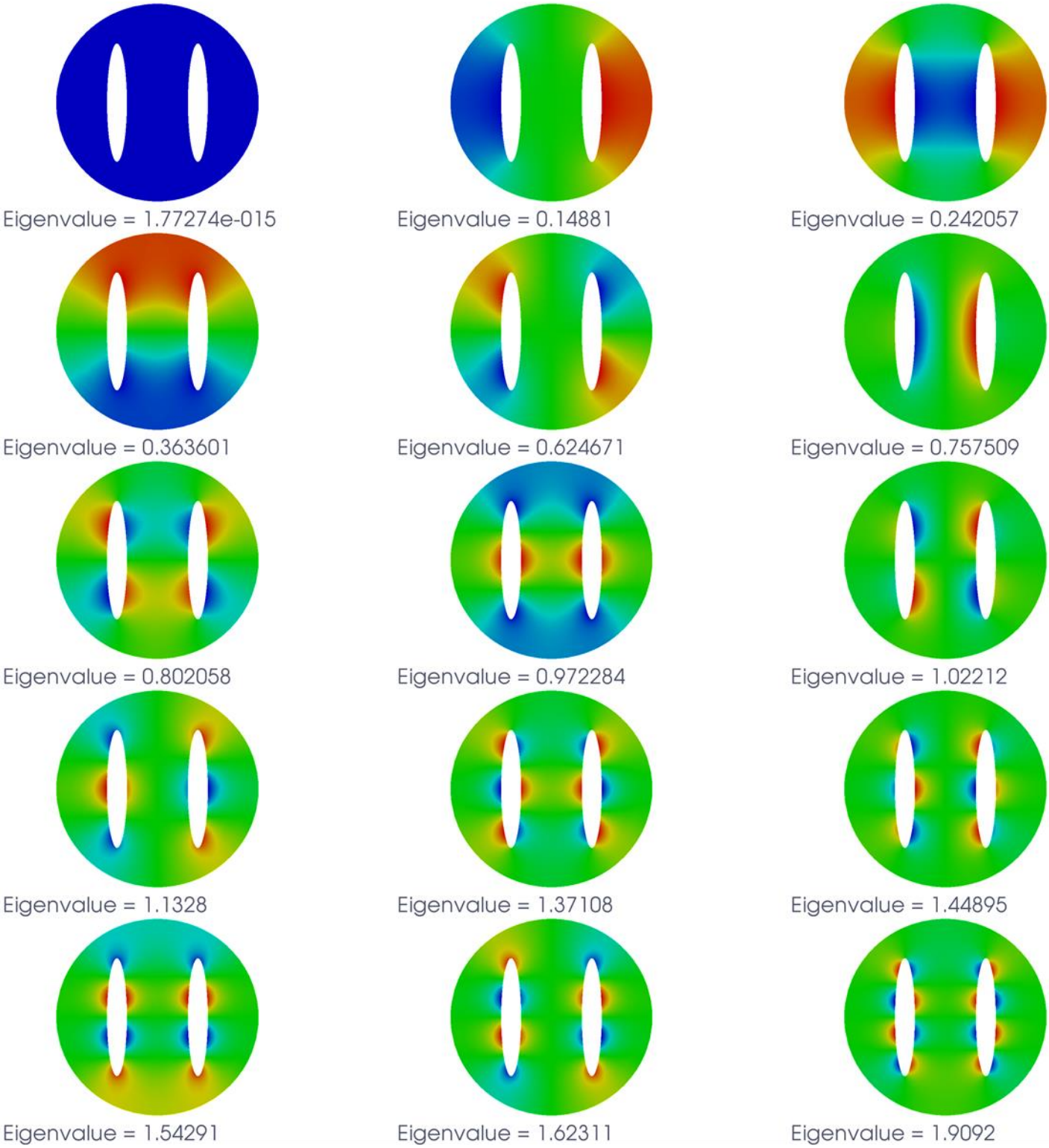


Figure 25: First 15 ordered Steklov-eigenfunctions (left to right) of the Electrostatics problem.

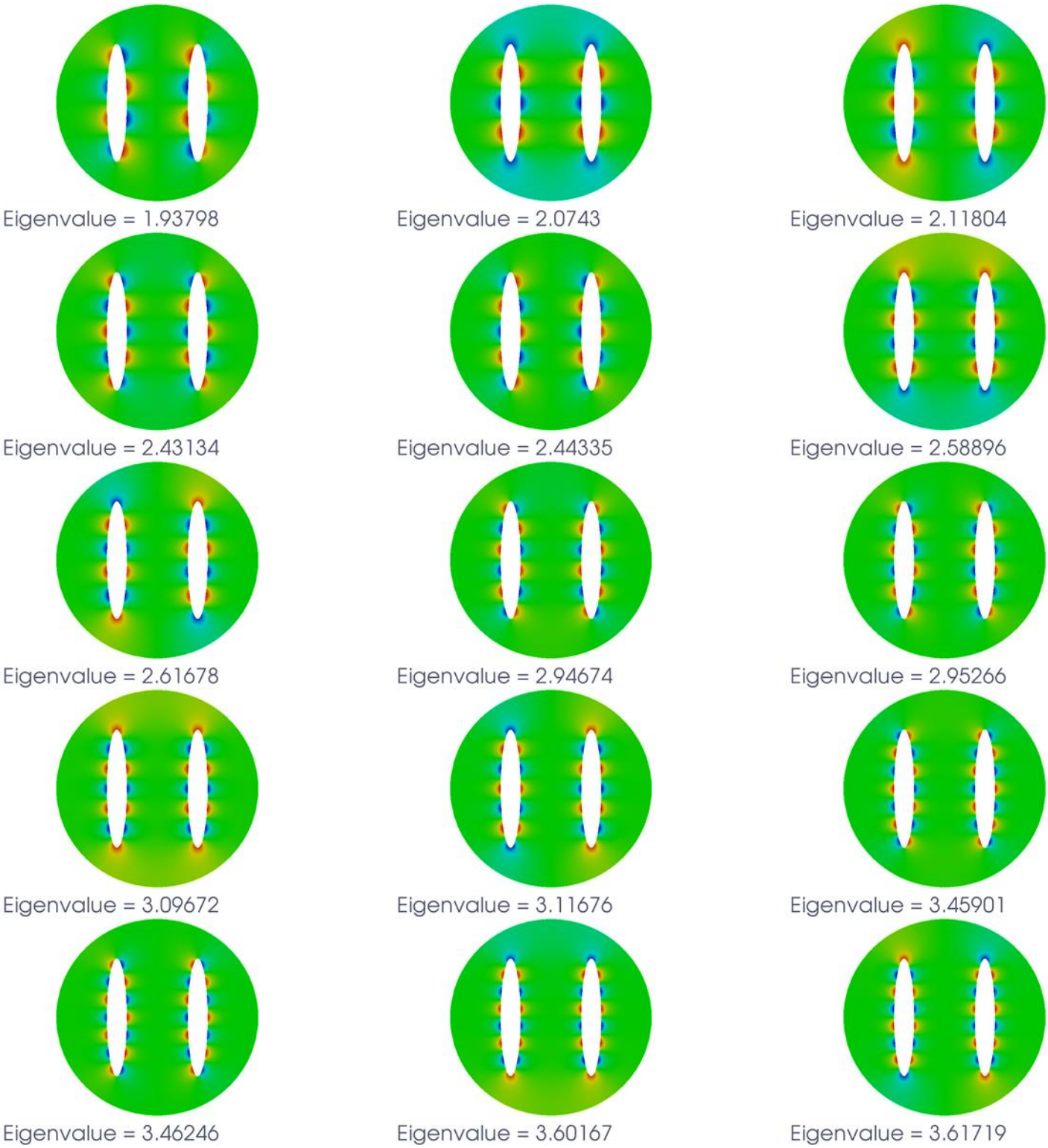


Figure 26: Next 15 ordered Steklov-eigenfunctions (left to right) of the Electrostatics problem. Notice how the eigenfunction becomes more localized near the boundary as the eigenvalue order increases.

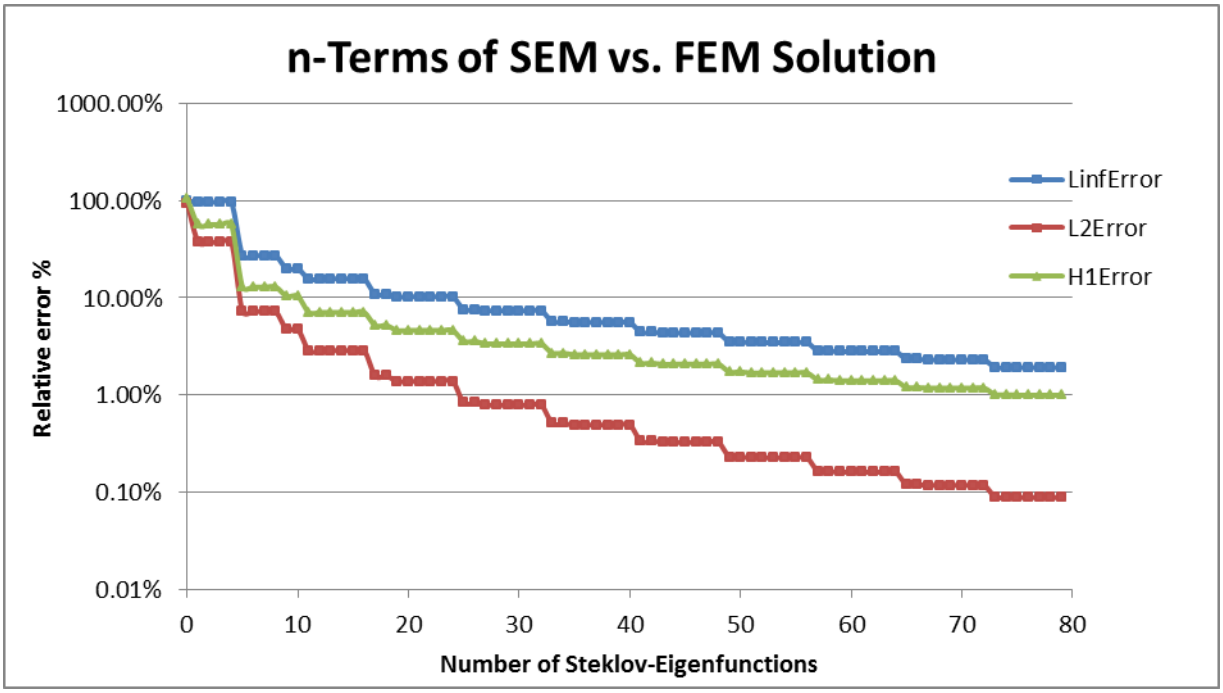


Figure 27: Relative errors for different norms comparing n-successive terms in the Steklov-expansion (x-axis) compared against solution produced by FEM (y-axis).

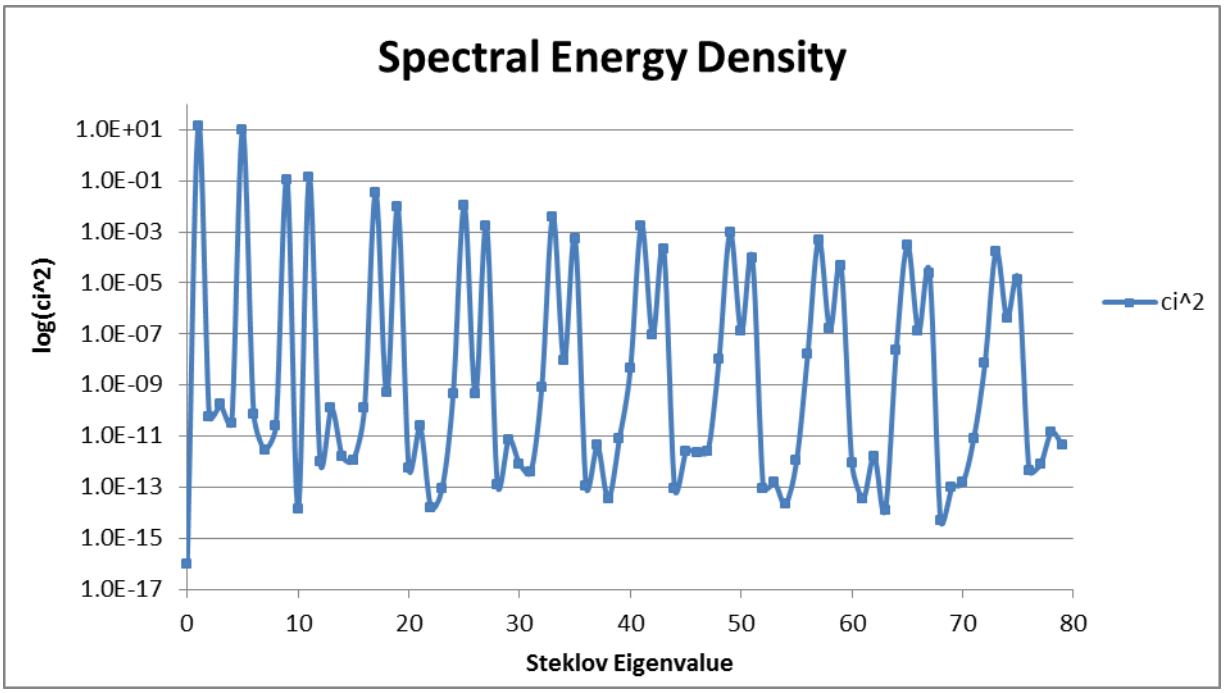


Figure 28: Spectral energy density given by $\langle s_n, g \rangle_{\Sigma}^2$.

	Steklov Eigenvalue	Relative L_∞-Error n SEM Terms Vs FEM Solution	Relative $L_2 - Error$ n SEM Terms Vs FEM Solution	Relative $H_1 - Error$ n SEM Terms Vs FEM Solution	n Coefficient in SEM Expansions
n	δ_n	$\frac{\|u_n^S - u_{FEM}^*\|_\infty}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^S - u_{FEM}^*\|_2}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^S - u_{FEM}^*\|_{1,2}}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\langle s_n, g\rangle_\Sigma$
0	1.77E-15	1	0.944751	1.05872	9.91E-09
1	0.14881	0.965159	0.374473	0.565685	3.82539
2	0.242057	0.965157	0.374473	0.565685	7.50E-06
3	0.363601	0.965157	0.374473	0.565685	-1.35E-05
4	0.624671	0.965157	0.374473	0.565685	5.68E-06
5	0.757509	0.272092	0.072126	0.129212	3.15407
6	0.802058	0.272092	0.072126	0.129212	8.67E-06
7	0.972284	0.272091	0.072126	0.129212	-1.65E-06
8	1.02212	0.272091	0.072126	0.129212	-5.02E-06
9	1.1328	0.198469	0.047145	0.10368	0.34115
10	1.37108	0.198469	0.047145	0.10368	-1.16E-07
11	1.44895	0.156388	0.028571	0.070684	0.366737
12	1.54291	0.156388	0.028571	0.070684	9.81E-07
13	1.62311	0.15639	0.028571	0.070684	1.12E-05
14	1.9092	0.156389	0.028571	0.070684	-1.29E-06
15	1.93798	0.15639	0.028571	0.070684	-1.08E-06
16	2.0743	0.15639	0.028571	0.070684	-1.12E-05
17	2.11804	0.108639	0.015692	0.051912	-0.19079
18	2.43134	0.108637	0.015692	0.051912	-2.21E-05
19	2.44335	0.102425	0.013792	0.04607	-0.09756
20	2.58896	0.102425	0.013792	0.04607	-7.42E-07
21	2.61678	0.102424	0.013792	0.04607	-4.92E-06
22	2.94674	0.102424	0.013792	0.04607	1.25E-07
23	2.95266	0.102424	0.013792	0.04607	2.93E-07
24	3.09672	0.10243	0.013792	0.04607	2.11E-05
25	3.11676	0.075853	0.00829	0.035694	0.102084
26	3.45901	0.075854	0.00829	0.035694	2.14E-05
27	3.46246	0.073645	0.007856	0.03357	0.042499
28	3.60167	0.073645	0.007856	0.03357	3.53E-07
29	3.61719	0.073644	0.007856	0.03357	2.74E-06

Table 5: Steklov eigenvalues for Electrostatics problem on a circular conductor. Relative errors in different norms captured for n-successive terms in the Steklov-expansion compared against solution produced by FEM.

(iii) Inviscid Fluid Flow between Parallel Plates

A simple example of a div-curl system is to consider an incompressible and irrotational planar flow of a fluid between two plates. The velocity field of an incompressible fluid satisfies $div\ v = 0$ and the irrotational assumption implies $curl\ v = 0$. A fluid of this type has negligible viscosity and is called inviscid. The geometry of the parallel plates is a rectangle, similar to problem (i). The top and bottom walls are impermeable plates enforced with a no-flux condition. The remaining left and right walls possess a flowing fluid measured in terms of a velocity flux, $(v.n_1)(y) = \mu_1(y)$ and $(v.n_2)(y) = \mu_2(y)$, such that the net flux across the boundary $\int_{\partial\Omega}\mu_1\ dy - \int_{\partial\Omega}\mu_2\ dy$ is zero. This problem fits the Neumann Harmonic problem discussed in section 4-(v). The problem takes the form:

Let $v(x)$ be the unknown fluid velocity between the plates in the planar region $\Omega = (-\mathrm{L}, \mathrm{L}) \times (-\mathrm{M}, \mathrm{M})$. There is a velocity potential $\varphi(x)$, $v = \nabla\varphi$, that satisfies

$$\Delta\varphi(x) = 0,\ \ x \in \Omega \tag{58}$$

subject to the Neumann boundary conditions

$$-\frac{\partial\varphi}{\partial x}(-L, y) = \mu_1(y) = (1 - (\frac{y}{M})^2)\,, \tag{59}$$

$$\frac{\partial\varphi}{\partial x}(L, y) = \mu_2(y) = (1 - (\frac{y}{M})^2)\,, for\ \ y \in [-M, M],$$

$$\frac{\partial\varphi}{\partial y}(x, M) = \frac{\partial\varphi}{\partial y}(x, -M) = 0,\ \ for\ x \in [-L, L]. \tag{60}$$

The equivalent mixed variational formulation is to find $\varphi \in H^1(\Omega)$ and $\lambda \in \mathbb{R}$ such that

$$\int_\Omega \nabla\varphi \cdot \nabla\chi\ \ dx + \int_\Omega \varphi\beta\ dx + \int_\Omega \lambda\chi\ dx = \int_{\partial\Omega}\mu\chi\ d\sigma, \text{ for all } \chi \in H^1(\Omega), \beta \in \mathbb{R}. \tag{61}$$

The associated Steklov-eigenvalues for this problem are similar to those found in the heat conduction problem (i). This demonstration does not delve into the details of comparing results with the analytical solution. See problem (i) for details.

```
macro Grad(u) [dx(u),dy(u)]
// Geometry - Solid Plate
real L = 2.0, M=1.0;  // L - value
border Left (t=0,1) {x=-L;y=M*(1.0-2.0*t);label=1;}  // Left barrier
border Top  (t=0,1) {x=L*(1.0-2.0*t);y=M;label=2;}   // Top wall
border Right(t=0,1) {x=L;y=M*(2.0*t-1.0);label=3;}   // Right barrier
border Bottom(t=0,1){x=L*(2.0*t-1.0);y=-M;label=2;}  // Bottom wall

// Construct a mesh
mesh Th = buildmesh(Left(50) + Top(50) + Right(50) + Bottom(50));

// Define inlet/outlet
func g1 = 1.0-(y/M)^2;
func g2 = 1.0-(y/M)^2;

// Finite element and functions
fespace Vh(Th,P2);
Vh uhFEM,uhSEM,vh;                          // Holds the final uhFEM
int eigCount=33, numEigs=3, count=0; // total number of eigenvalues
real[int] ev(eigCount);                     // Eigenvalues
Vh [int]   eV(eigCount),eVold(eigCount),pSum(eigCount);  // Eigenvectors

// Steklov - Eigenvalue problem in variational form
real L2Errorsq = 1.0E8, adaptErr = 1.0e-2, shift = 1.0e-2;
varf vA(u,v) = int1d(Th,1,3)(-shift*u* v)+int2d(Th)(dx(u)*dx(v)+dy(u)*dy(v));
varf vB(u,v) = int1d(Th,1,3)( u*v) ;
while(L2Errorsq > 3.0E-2)
{
  if (count > 0 && count < 3) // Adaptation step
  {
     Vh fAdapt = 0;
     for(int i=1;i<numEigs;i++)
        fAdapt = fAdapt +  abs(eV[i]);
     Th = adaptmesh(Th,fAdapt,err=adaptErr);
     adaptErr/=2.0;
  } else if (count > 0) Th = splitmesh(Th,2);

  // Get resulting matrices Ax=lBx
  matrix A = vA(Vh,Vh,solver = sparsesolver);
  matrix B = vB(Vh,Vh);
```

```
  // Solve Ax=lBx
  numEigs = EigenValue(A,B,sym=true,sigma=0,value=ev,vector=eV);
  numEigs = min(eigCount,numEigs);
  L2Errorsq = 0.0;
  for(int i=1;i<numEigs;i++)
  {
     Vh ssDiff = abs(abs(eV[i])-abs(eVold[i]));
     L2Errorsq = max(L2Errorsq,int2d(Th)(ssDiff*ssDiff));
     eVold[i] = eV[i];
  }
  count++;
}
// Construct solution
real[int] c(numEigs);
for(int i=1;i<numEigs;i++)
{
  c[i] =(int1d(Th,1)(g1*eV[i]) + int1d(Th,3)(g2*eV[i]))/ ev[i];
  uhSEM = uhSEM + c[i] * eV[i];
  pSum[i] = uhSEM;
}
// Solve mixed DN FEM problem
varf va(uh,vh) = int2d(Th)(Grad(uh)'*Grad(vh));
varf vb(uh,vh) = int2d(Th)(vh);
varf vL(uh,vh) = -int1d(Th,1)(g1*vh) +  // Left wall
                  int1d(Th,3)(g2*vh);  // right wall
// Capture # of degrees of freedom to construct mixed matrix
int n1 = Vh.ndof+1;
// Construct mixed matrix to solve mixed FEM problem
//   A*uh + B*l = b
//   B*uh=0
matrix   A = va(Vh,Vh);   // Stiffness Block matrix A
real[int] b = vL(0,Vh);   // Solution vector
real[int] B = vb(0,Vh);   // Constraint block matrix
real[int] bb(n1),  // Modified solution vector for mixed problem
          xx(n1),  // Solution vector
             l(1);   // lagrange multiplier
matrix   AA = [[A,B],[B',0]];  // Mixed block matrix
set(AA,solver=sparsesolver); // Set solver type on matrix
bb = [b,0];     // Construct right hand side vector
xx = (AA^-1)*bb; // Get solution vector
[uhFEM[],l] = xx;   // Part of vector has solution, other has lagrange multiplier
// Plot solutions
plot(uhSEM,value=true,fill=true,ColorScheme=2);
plot(uhFEM,value=true,fill=true,ColorScheme=2);
```

Code Snippet 12: Code to solve Neumann BVP with FEM and adapted SEM.

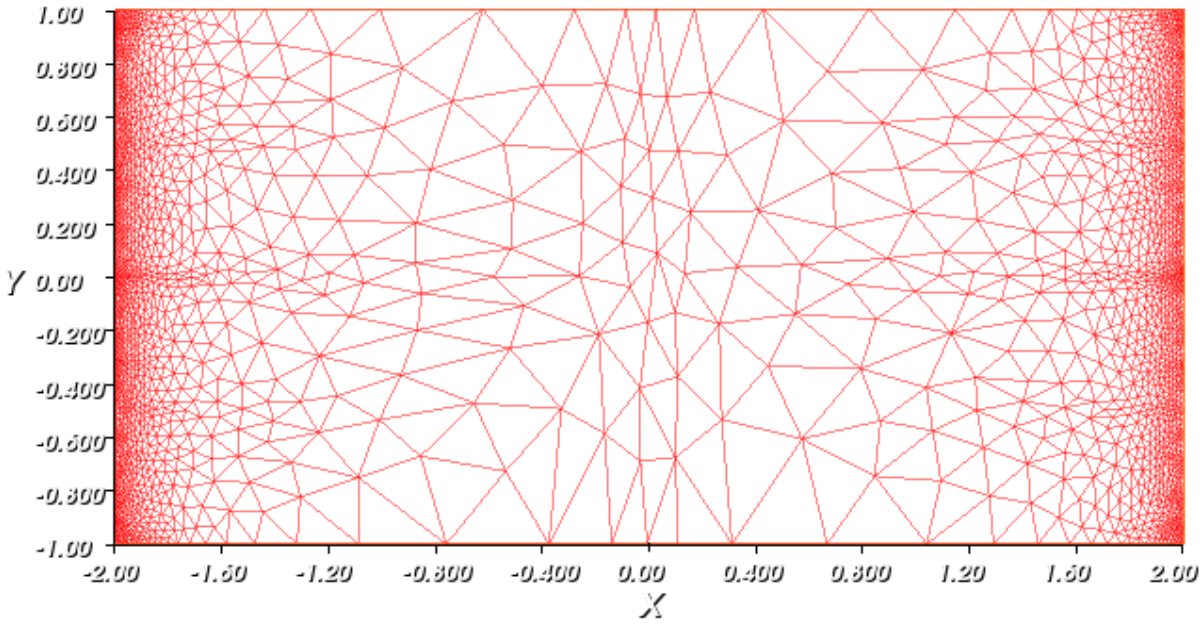


Figure 29: Adapted mesh for SEM method

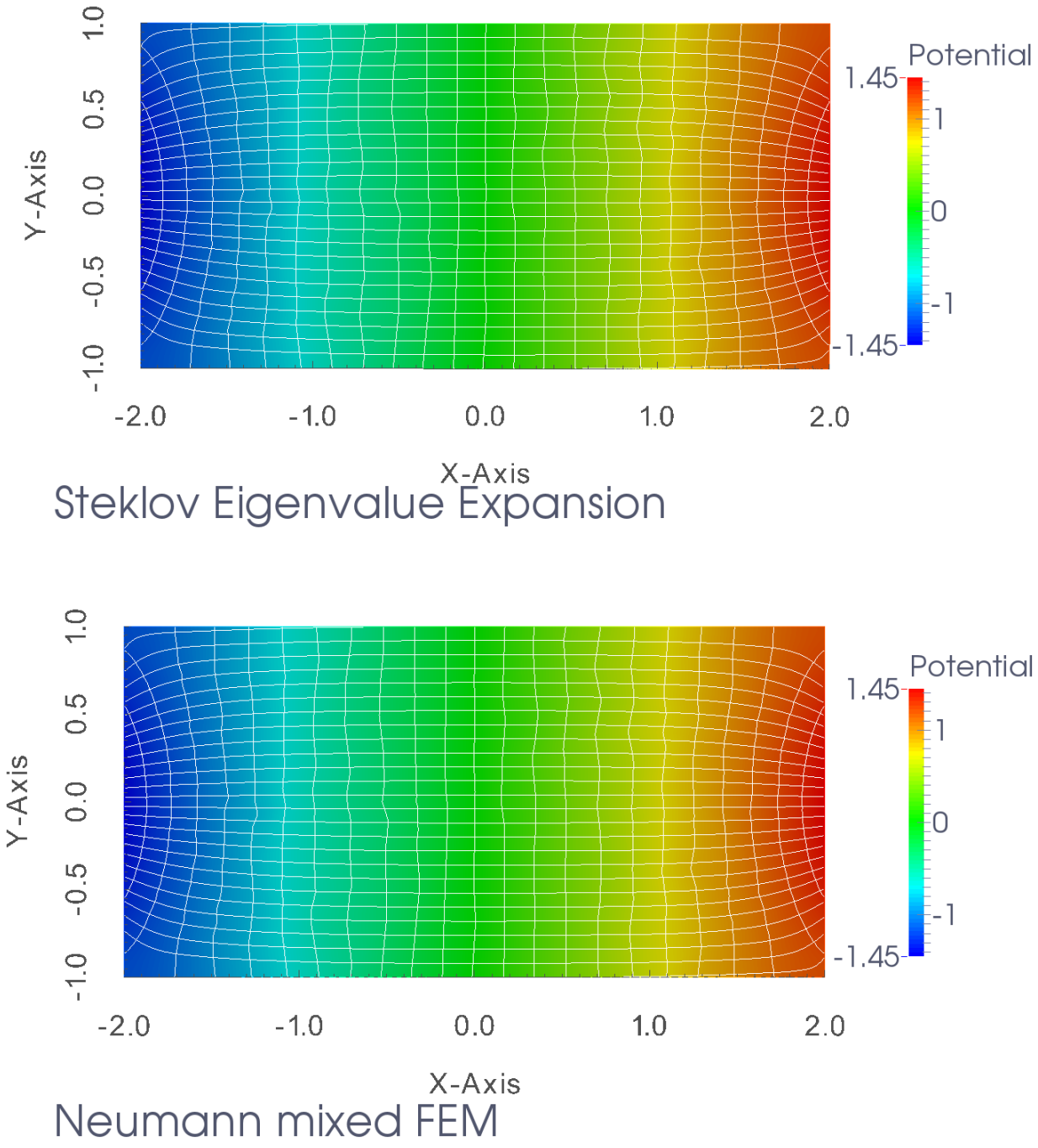


Figure 30: Solutions from both SEM & FEM procedures. Horizontal particle streamlines and vertical potential lines are overlayed.

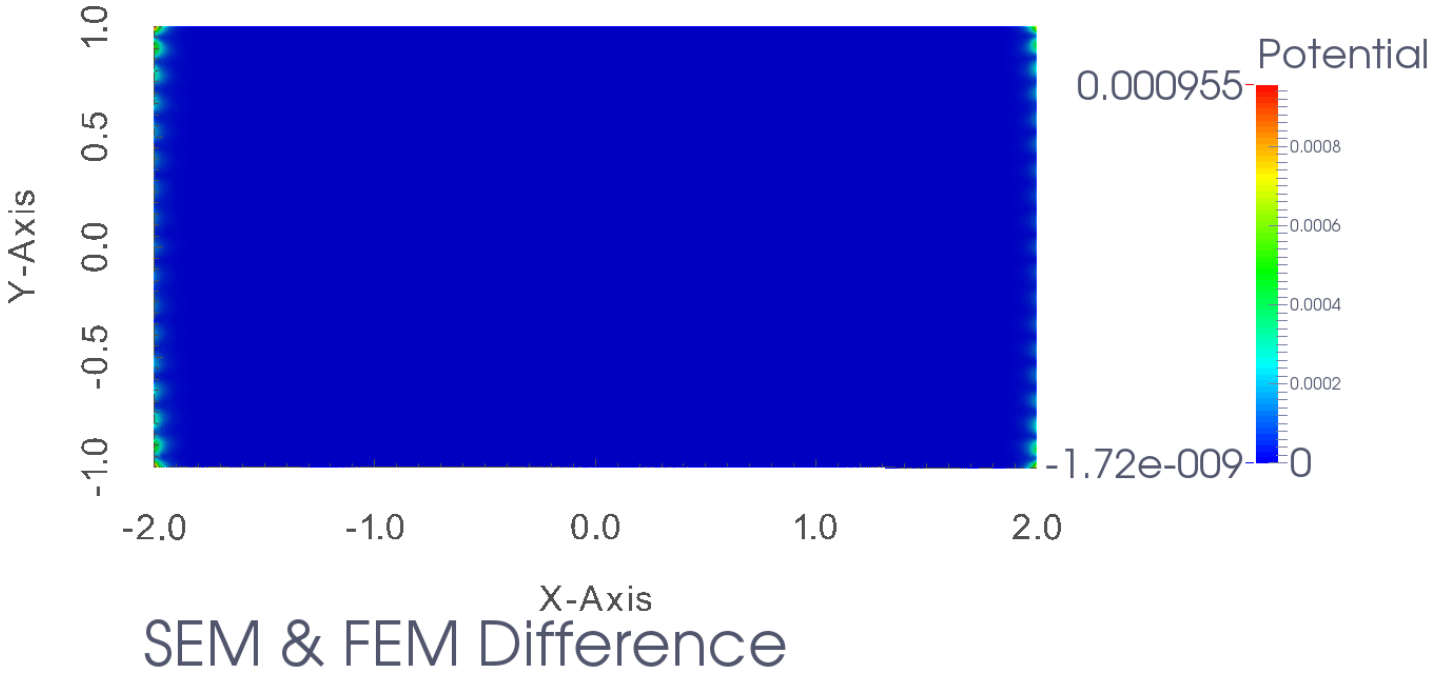


Figure 31: Absolute difference of SEM & FEM solutions. Some difference is revealed on the boundary.

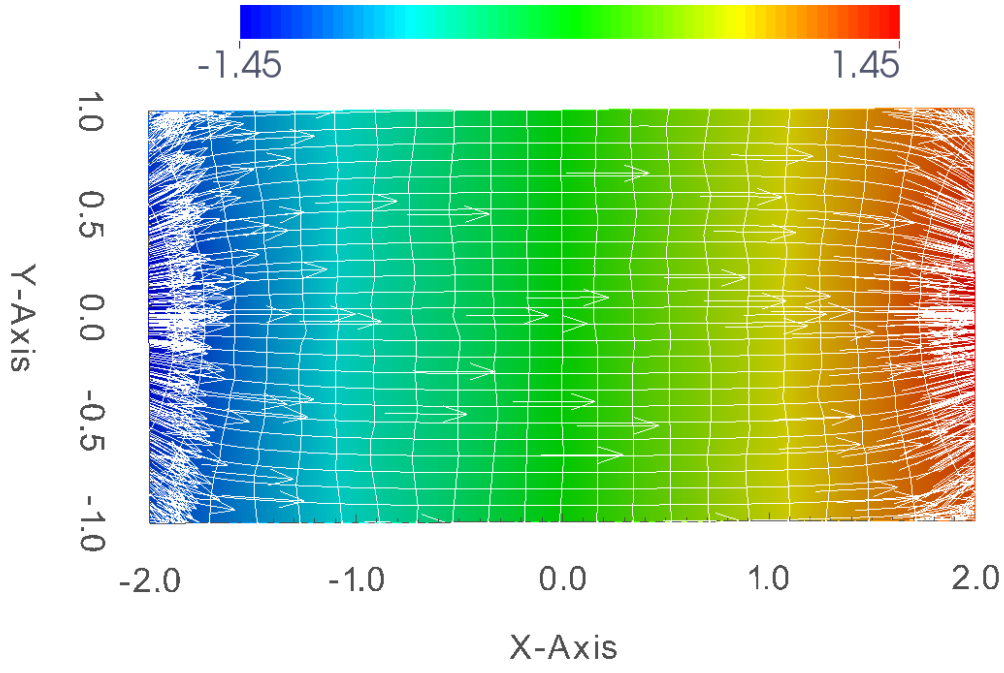


Figure 32: Velocity field showing flow of fluid between parallel plates.

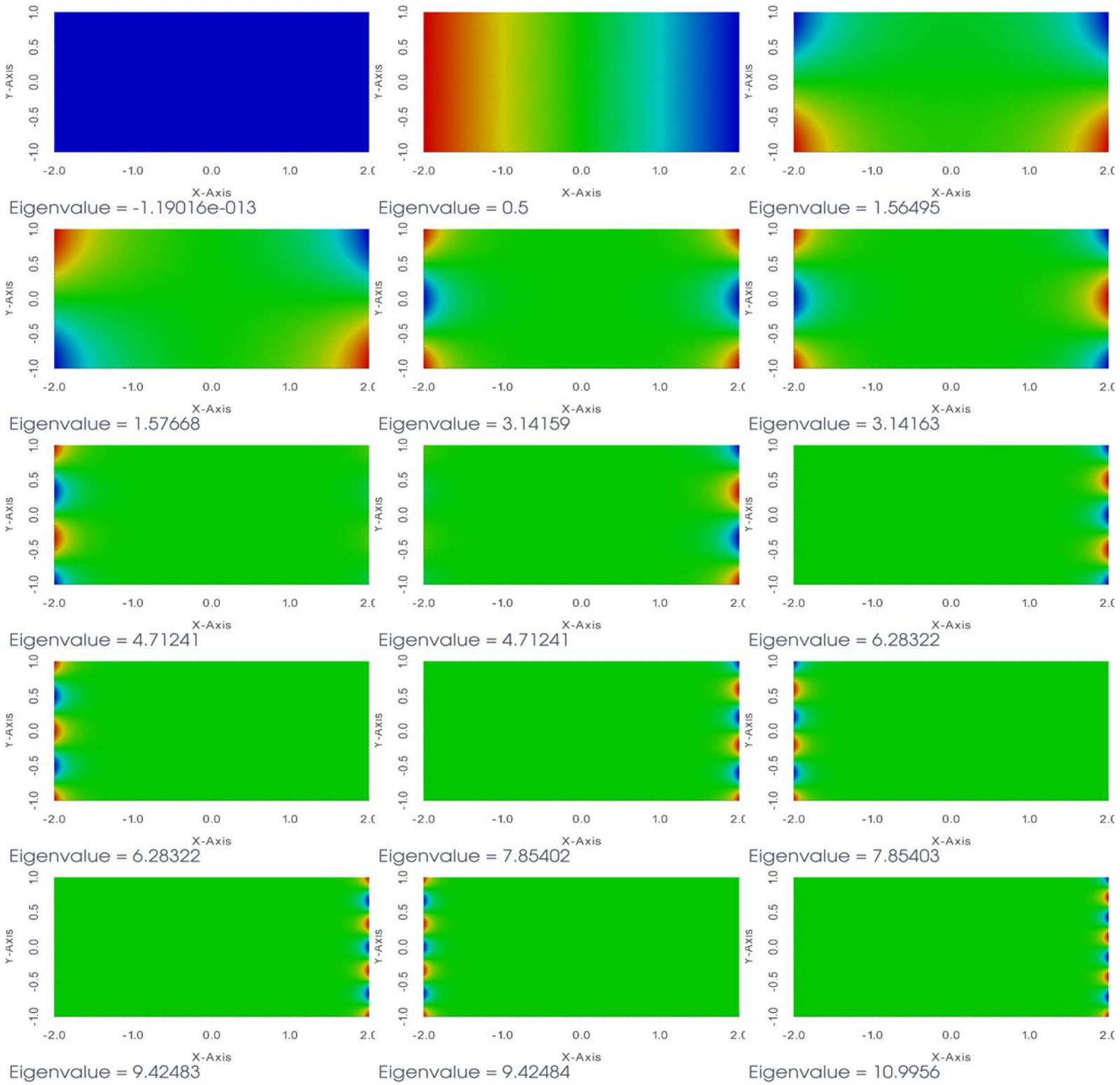


Figure 33: First 15 ordered Steklov-eigenfunctions (left to right) of the Inviscid Flow between Parallel Plates problem.

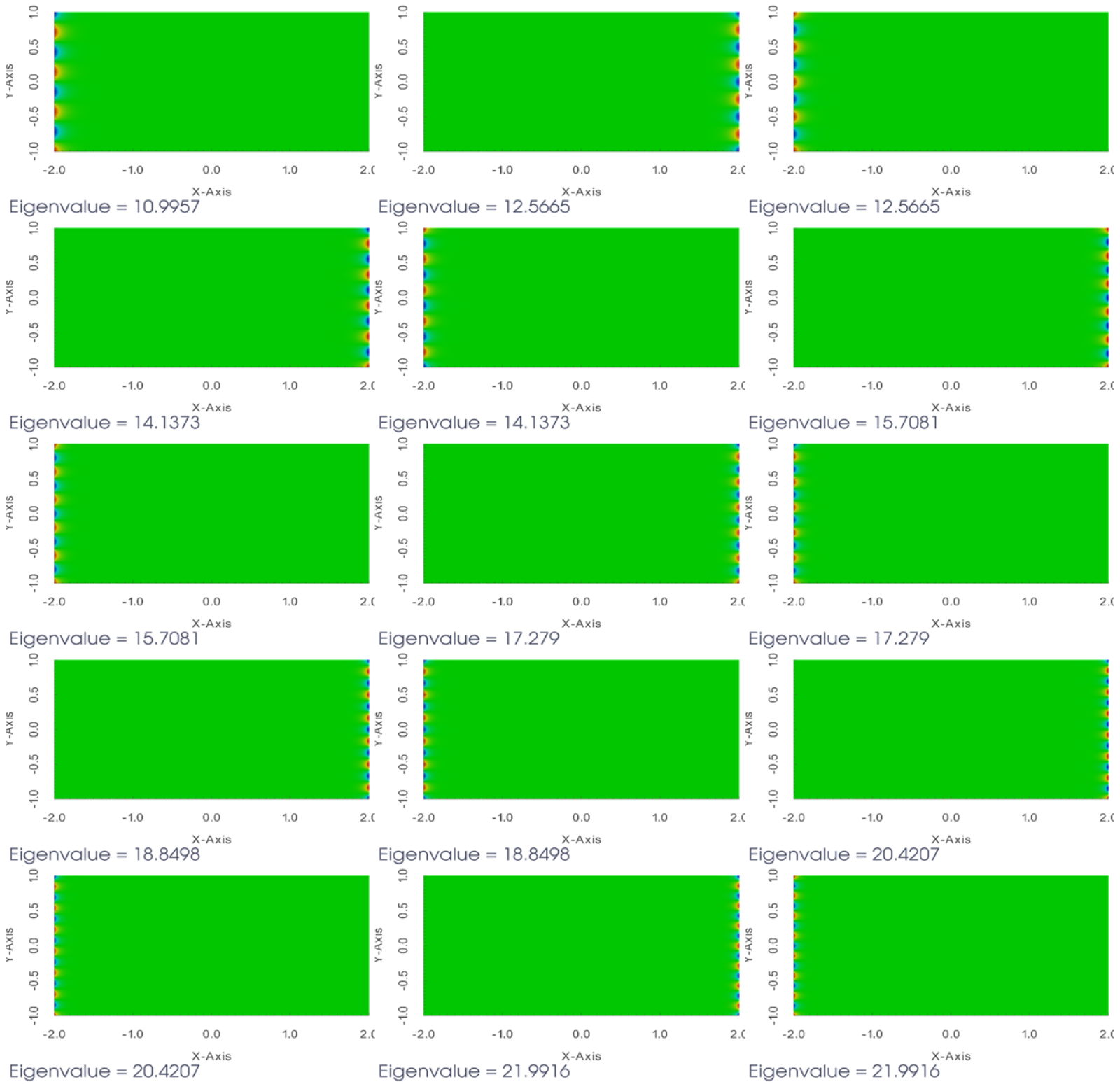


Figure 34: Next 15 ordered Steklov-eigenfunctions (left to right) of the Inviscid Flow between Parallel Plates problem. Notice how the eigenfunction becomes more localized near the boundary as the eigenvalue order increases.

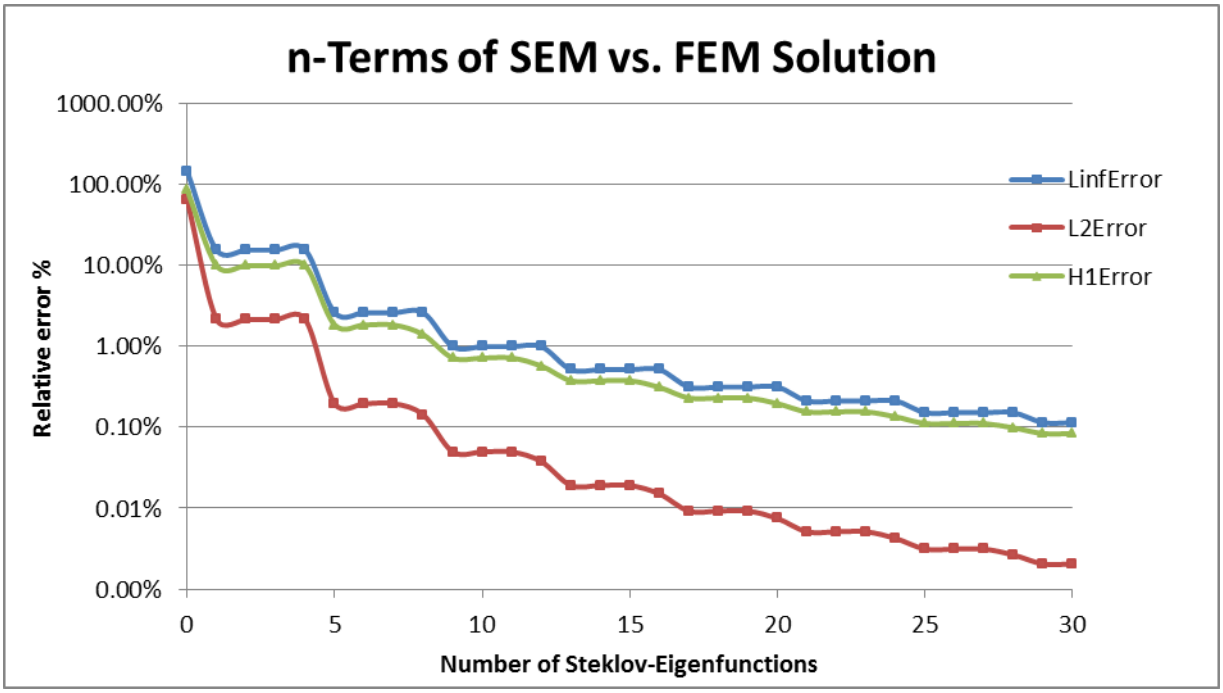


Figure 35: Relative errors for different norms comparing n-successive terms in the Steklov-expansion (x-axis) compared against solution produced by FEM (y-axis).

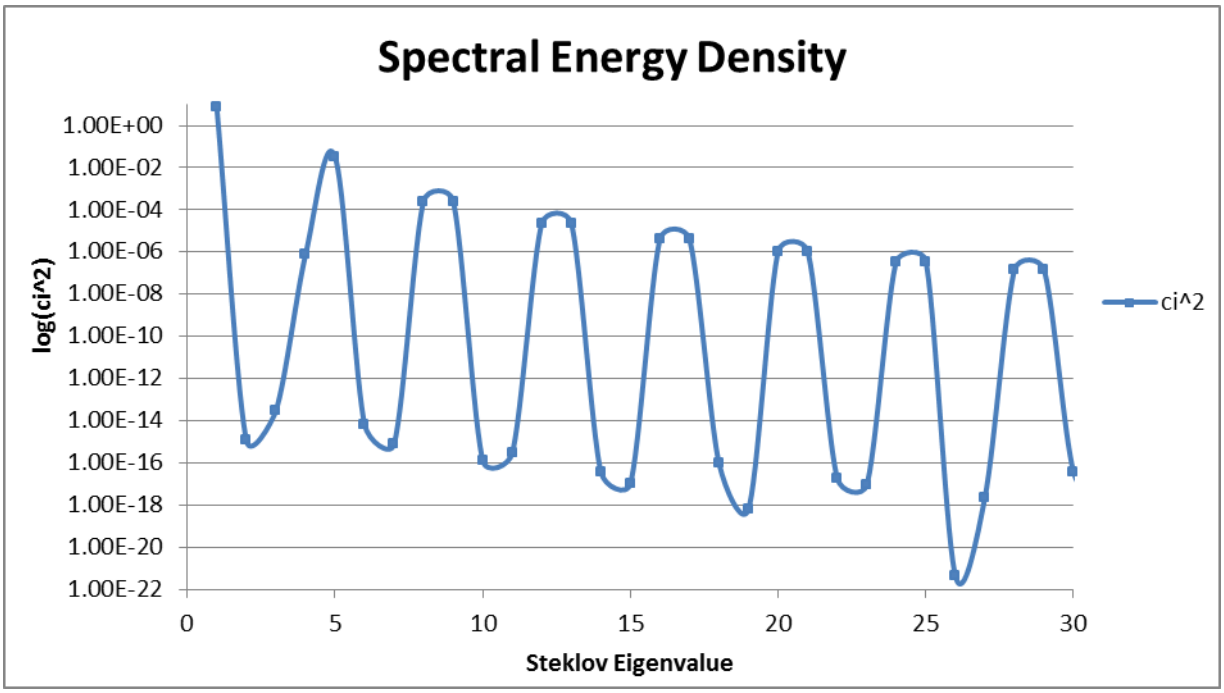


Figure 36: Spectral energy density given by $\langle s_n, g \rangle_2^2$.

	Steklov Eigenvalue	Relative L_∞-Error n SEM Terms Vs FEM Solution	Relative $L_2 - Error$ n SEM Terms Vs FEM Solution	Relative $H_1 - Error$ n SEM Terms Vs FEM Solution	n Coefficient in SEM Expansions
n	δ_n	$\frac{\|u_n^s - u_{FEM}^*\|_\infty}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^s - u_{FEM}^*\|_2}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^s - u_{FEM}^*\|_{1,2}}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\langle s_n, g\rangle_\Sigma$
0	-1.19E-13	1.44964	0.642496	0.85352	0
1	0.5	0.155071	0.021554	0.099294	-2.66667
2	1.56495	0.155071	0.021554	0.099294	3.44E-08
3	1.57668	0.155071	0.021554	0.099294	-1.73E-07
4	3.14159	0.155673	0.021553	0.099296	0.000855
5	3.14163	0.026067	0.001961	0.018392	0.182438
6	4.71241	0.026067	0.001961	0.018392	8.17E-08
7	4.71241	0.026067	0.001961	0.018392	2.78E-08
8	6.28322	0.026068	0.001431	0.01396	0.016125
9	6.28322	0.009941	0.000497	0.007188	0.016127
10	7.85402	0.009941	0.000497	0.007188	1.14E-08
11	7.85403	0.009941	0.000497	0.007188	-1.71E-08
12	9.42483	0.009941	0.000377	0.005737	-0.00478
13	9.42484	0.005163	0.000191	0.003764	0.004778
14	10.9956	0.005163	0.000191	0.003764	6.11E-09
15	10.9957	0.005163	0.000191	0.003764	3.22E-09
16	12.5665	0.005163	0.00015	0.003121	0.002016
17	12.5665	0.003147	9.23E-05	0.002304	0.002016
18	14.1373	0.003147	9.23E-05	0.002304	-9.68E-09
19	14.1373	0.003147	9.23E-05	0.002304	-8.22E-10
20	15.7081	0.003147	7.46E-05	0.001964	-0.00103
21	15.7081	0.002115	5.12E-05	0.001552	0.001032
22	17.279	0.002115	5.12E-05	0.001552	-4.33E-09
23	17.279	0.002115	5.12E-05	0.001552	3.05E-09
24	18.8498	0.002115	4.24E-05	0.001351	0.000597
25	18.8498	0.001518	3.13E-05	0.001114	-0.0006
26	20.4207	0.001518	3.13E-05	0.001114	2.08E-11
27	20.4207	0.001518	3.13E-05	0.001114	-1.53E-09
28	21.9916	0.001518	2.64E-05	0.000986	0.000376
29	21.9916	0.001142	2.05E-05	0.000838	0.000376

Table 6: Steklov Eigenvalues for the inviscid flow between parallel plates. Relative errors in different norms captured for n-successive terms in the Steklov-expansion compared against solution produced by FEM.

(iv) Inviscid Fluid Flow (Contraction-Expansion)

Example (iii) resulted in a well-behaved field of parallel lines across the region. The field near the left and right walls is slightly perturbed from parallel lines due to the boundary conditions. The next two examples continue to assume inviscid incompressible planar flow. The geometry in this example has a change in area called an expansion that opens the flow from a constricted zone to a less constricted zone. This flow field is more interesting than the flow between the parallel plates example. The top and bottom walls are impermeable plates with a no-flux condition. The walls will open up at a point along the x-axis. The remaining left and right walls continue to contain a velocity flux, $(v.n_1)(y) = \mu_1(y)$ and $(v.n_2)(y) = \mu_2(y)$, such that the net flux across the boundary $\int_{\partial\Omega}\mu_1\, dy - \int_{\partial\Omega}\mu_2\, dy$ is zero. The problem takes the form:

Let $v(x)$ be the unknown fluid velocity between the expanded plates in the planar region $\Omega = (-\mathrm{L}, 0) \times (-\mathrm{M}_L, \mathrm{M}_L) \cup (0, \mathrm{L})) \times (-\mathrm{M}_R, \mathrm{M}_R)$ with $M_L \leq M_R$. There is a velocity potential $\varphi(x)$, $v = \nabla\varphi$, that satisfies

$$\Delta\varphi(x) = 0, \;\; x \in \Omega \tag{62}$$

subject to the Neumann boundary conditions

$$-\frac{\partial\varphi}{\partial x}(-L, y) = \mu_1(y) = \left(1 - \left(\frac{y}{M_L}\right)^2\right), for \;\; y \in [-\mathrm{M}_L, \mathrm{M}_L]\,, \tag{63}$$

$$\frac{\partial\varphi}{\partial x}(L, y) = \mu_2(y) = \left(1 - \left(\frac{y}{\mathrm{M}_R}\right)^2\right) M_L/M_R\,, for \;\; y \in [-\mathrm{M}_R, \mathrm{M}_R],$$

$$\frac{\partial\varphi}{\partial y}(x, M_L) = \frac{\partial\varphi}{\partial y}(x, -M_L) = 0, \;\; for\, x \in [-L, 0], \tag{64}$$

$$\frac{\partial\varphi}{\partial y}(x, M_R) = \frac{\partial\varphi}{\partial y}(x, -M_R) = 0, \;\; for\, x \in [0, L], \tag{65}$$

$$\frac{\partial\varphi}{\partial x}(0, y) = \frac{\partial\varphi}{\partial x}(0, -y) = 0, \;\; for\, y \in [M_L, M_R], \tag{66}$$

The equivalent mixed variational formulation is to find $\varphi \in H^1(\Omega)$ and $\lambda \in \mathbb{R}$ such that

$$\int_\Omega \nabla\varphi \cdot \nabla\chi \;\, dx + \int_\Omega \varphi\beta \, dx + \int_\Omega \lambda\chi \, dx = \int_{\partial\Omega} \mu\chi \, d\sigma, \text{ for all } \chi \in H^1(\Omega), \beta \in \mathbb{R}. \tag{67}$$

```
macro Grad(u) [dx(u),dy(u)]
// Geometry Parallel plates
real L = 8.0;  // L - value
real M = 1.0;
// Rectangle geometry
border Left(t=0,1){x=-L;y=M*(2.0*t-1.0);label=1;}    // Left wall
border Top1(t=0,1){x=L*(t-1.0);y=M;label=2;}        // Lower Top  wall
border Wall1(t=0,1){x=0.0;y=M+4.0*t;label=2;}         // Wall
border Top2(t=0,1){x=L*t;y=5.0*M;label=2;}            // Upper top wall
border Right(t=0,1){x=L;y=-5.0*M + 10*M*t;label=3;}    // Right wall
border Bott1(t=0,1){x=L*(t-1.0);y=-M;label=2;}  // Bottom wall
border Bott2(t=0,1){x=L*t;y=-5*M;label=2;}  // Bottom wall
border Wall2(t=0,1){x=0.0;y=-M-4.0*t;label=2;}
// Coarse rough uniform rectanglur mesh
real S = 33;
mesh Th = buildmesh(Left(-S) + Top1(-S) + Wall1(-S) + Top2(-S) + Right(S) + Bott1(S) + Bott2(S) + Wall2(S));
// Define inlet/outlet
func g1 = 1-(y/M)^2;
func g2 = (1-(y/(5*M))^2)/5;
// Finite element and functions
fespace Vh(Th,P2);
Vh uhFEM,uhSEM,vh;                          // Holds the final uhFEM
int eigCount=33, numEigs=33, count=0; // total number of eigenvalues
real[int] ev(eigCount);                     // Eigenvalues
Vh [int]   eV(eigCount),eVold(eigCount),pSum(eigCount);  // Eigenvectors
// Steklov - Eigenvalue problem in variational form
real L2Errorsq = 1.0E8, adaptErr = 1.0e-2, shift = 0;
varf vA(u,v) = int1d(Th, 1,3)(-shift* u*v)+int2d(Th)(dx(u)*dx(v)+dy(u)*dy(v));
varf vB(u,v) = int1d(Th, 1,3)( u*v);
while(L2Errorsq > 5.0E-1)
{
  if (count > 0 && count < 5) // Adaptation step
  {
     Vh fAdapt = 0;
     for(int i=1;i<numEigs;i++)
        fAdapt = fAdapt +  abs(eV[i]);
     Th = adaptmesh(Th,fAdapt,err=adaptErr);
     adaptErr/=2.0;
  } else if (count > 0) Th = splitmesh(Th,2);
  // Get resulting matrices Ax=lBx
  matrix A = vA(Vh,Vh,solver = sparsesolver);
  matrix B = vB(Vh,Vh);
```

```
  // Solve Ax=lBx
  numEigs = EigenValue(A,B,sym=true,sigma=shift,value=ev,vector=eV);
  numEigs = min(eigCount,numEigs);
  L2Errorsq = 0.0;
  for(int i=0;i<numEigs;i++)
  {
     Vh ssDiff = abs(abs(eV[i])-abs(eVold[i]));
     L2Errorsq = max(L2Errorsq,int2d(Th)(ssDiff*ssDiff));
     eVold[i] = eV[i];
  }  count++;
}
// Construct solution
real[int] c(numEigs);
for(int i=1;i<numEigs;i++)
{
  c[i] =(int1d(Th, 1)(g1*eV[i]) + int1d(Th, 3)(g2*eV[i]))/ ev[i];
  uhSEM = uhSEM + c[i] * eV[i];
  pSum[i] = uhSEM;
}
// Solve mixed DN FEM problem
varf va(uh,vh) = int2d(Th)(Grad(uh)'*Grad(vh));
varf vb(uh,vh) = int2d(Th)(vh);
varf vL(uh,vh) = -int1d(Th,1)(g1*vh) +  int1d(Th,3)(g2*vh);  // right wall
// Capture # of degrees of freedom to construct mixed matrix
int n1 = Vh.ndof+1;
// Construct mixed matrix to solve mixed FEM problem
//    A*uh + B*l = b
//    B*uh=0
matrix   A = va(Vh,Vh);   // Stiffness Block matrix A
real[int] b = vL(0,Vh);    // Solution vector
real[int] B = vb(0,Vh);    // Constraint block matrix
real[int] bb(n1),   xx(n1), l(1);   // Modified sol vector for mixed problem, sol vector, lagrange multiplier
matrix   AA = [[A,B],[B',0]];  // Mixed block matrix
set(AA,solver=sparsesolver); // Set solver type on matrix
bb = [b,0];      // Construct right hand side vector
xx = (AA^-1)*bb; // Get solution vector
[uhFEM[],l] = xx;   // Part of vector has solution, other has Lagrange multiplier
// Plot solutions
plot(uhSEM,value=true,fill=true,ColorScheme=2);
plot(uhFEM,value=true,fill=true,ColorScheme=2);
```

Code Snippet 13: Code to solve Neumann BVP with FEM and adapted SEM.

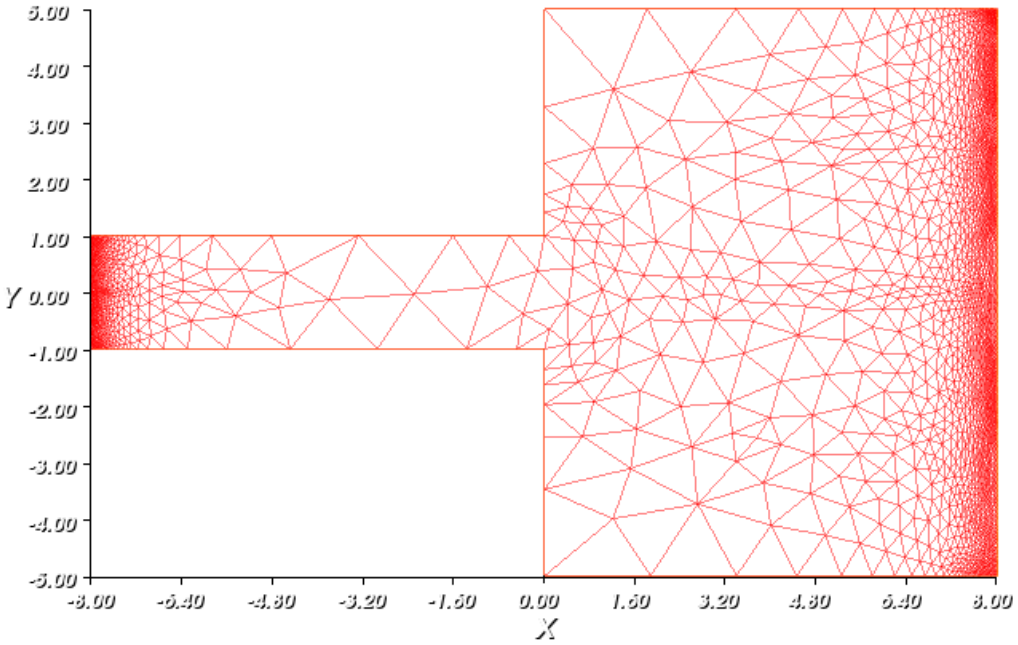


Figure 37: Adapted mesh for SEM method.

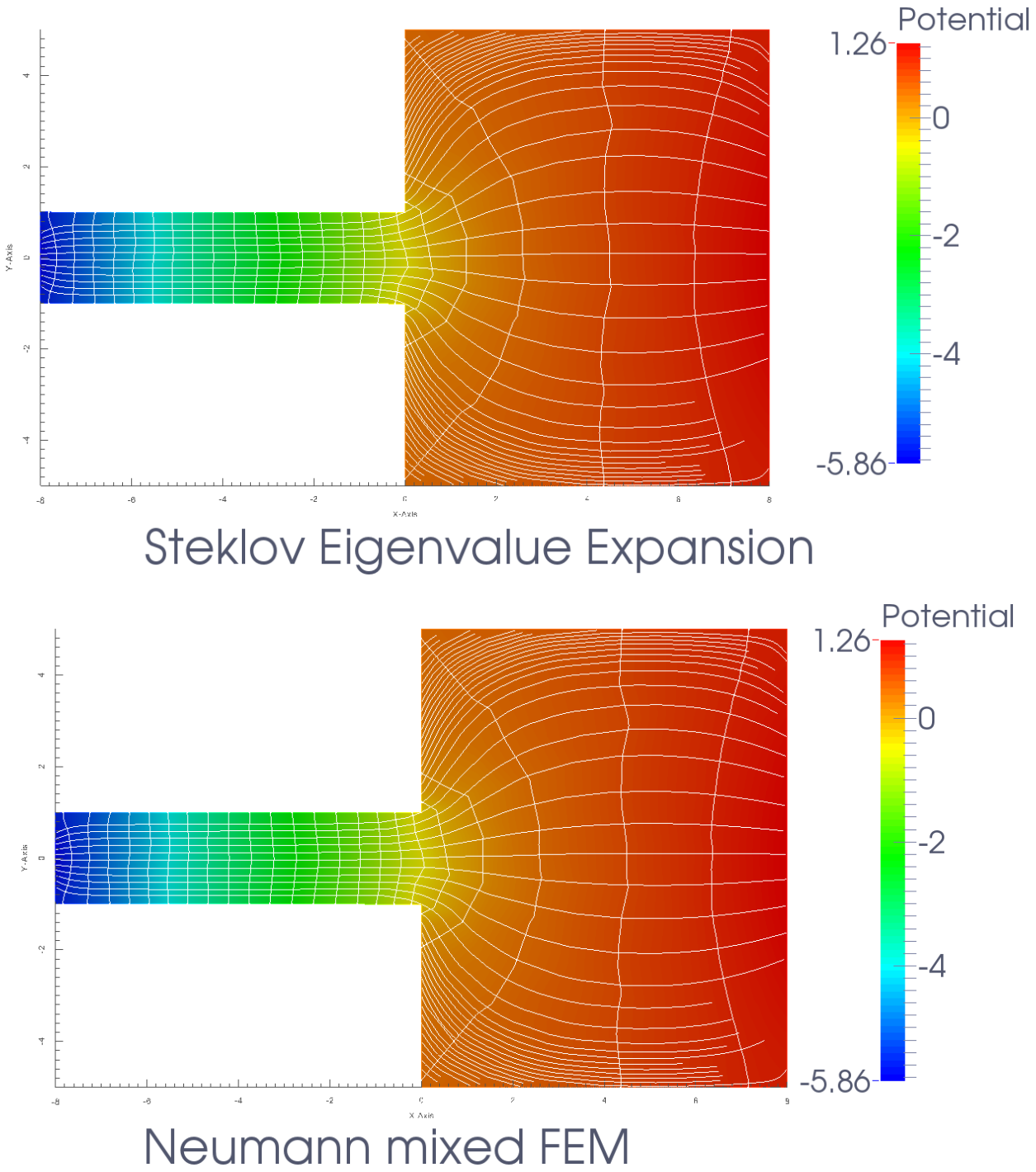


Figure 38: Solutions from both SEM & FEM procedures. Horizontal particle streamlines and vertical potential lines are overlayed.

Figure 39: Absolute difference of SEM & FEM solutions. Some difference is revealed on the left boundary.

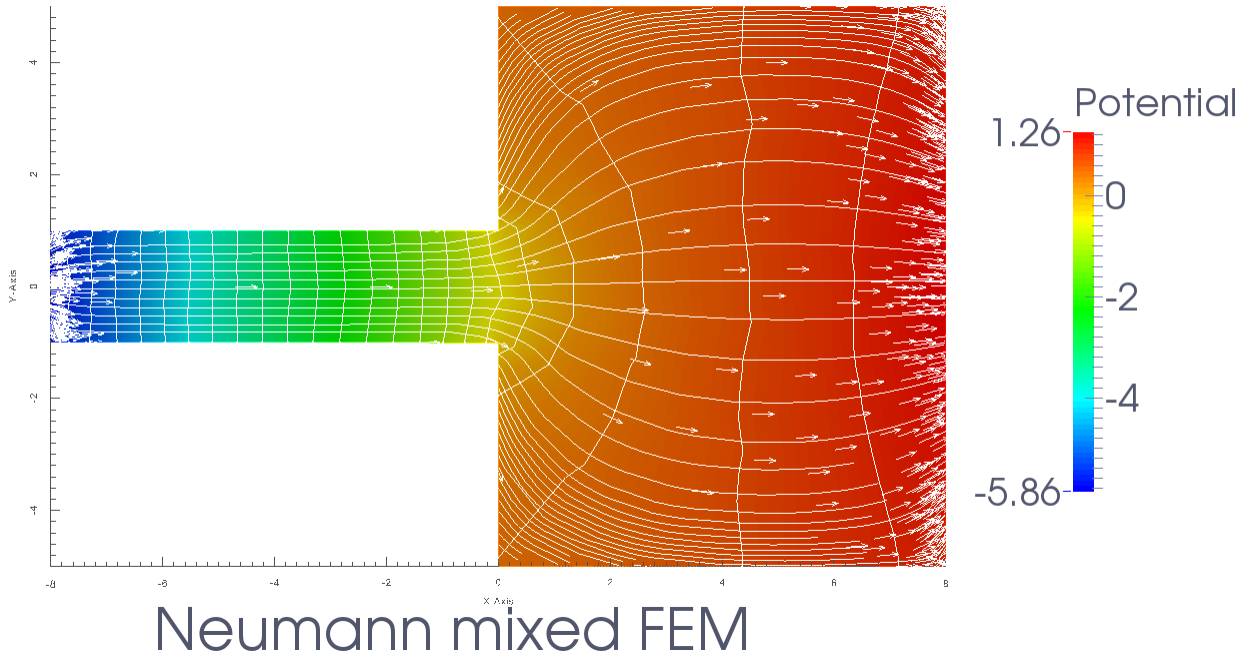


Figure 40: Vector field showing flow of fluid between parallel plates going through an expansion.

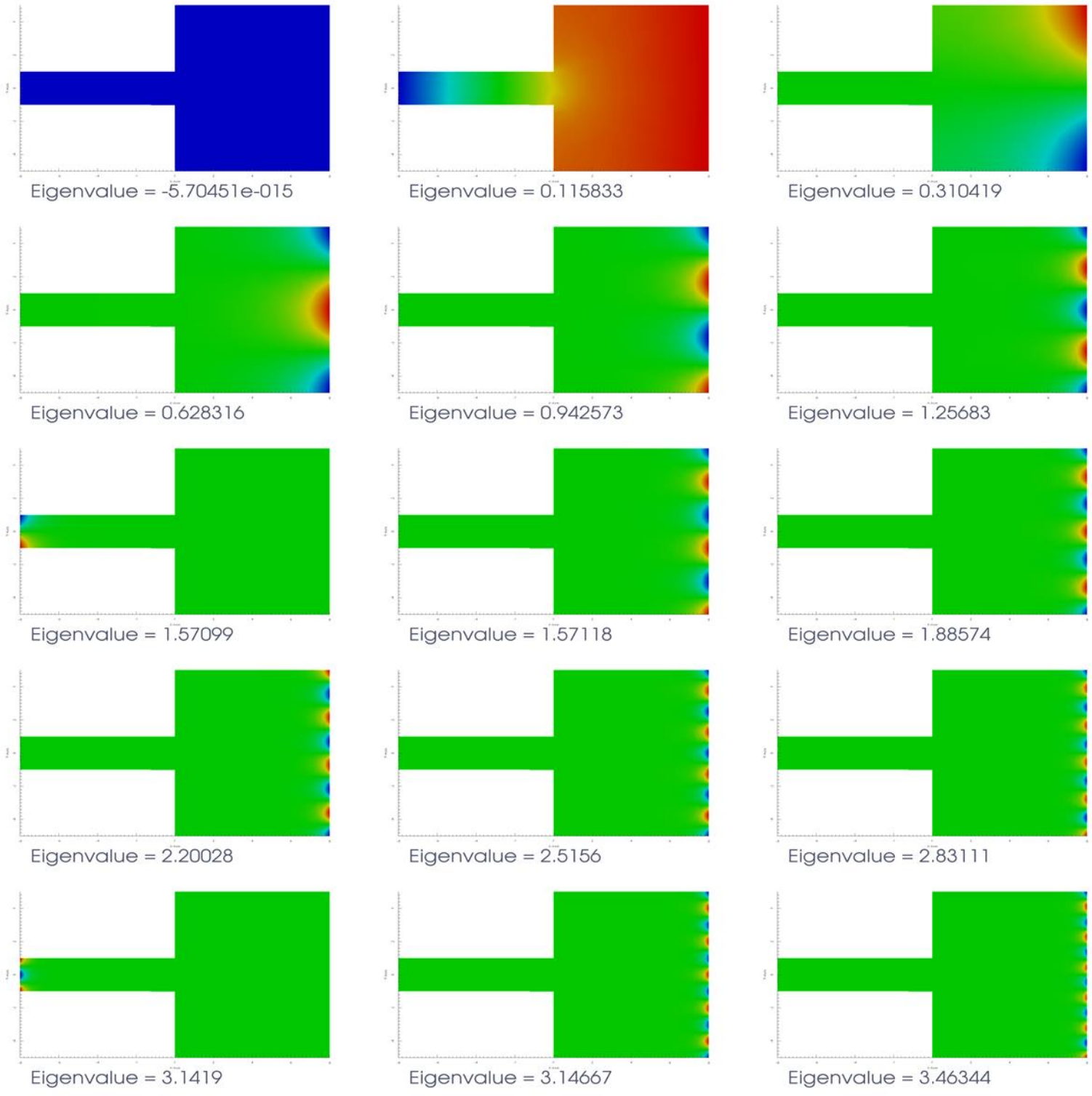


Figure 41: First 15 ordered Steklov-eigenfunctions (left to right) of the con.exp. between parallel plates problem.

Figure 42: Next 15 ordered Steklov-eigenfunctions (left to right) of the con.exp. beteween parallel plates problem.

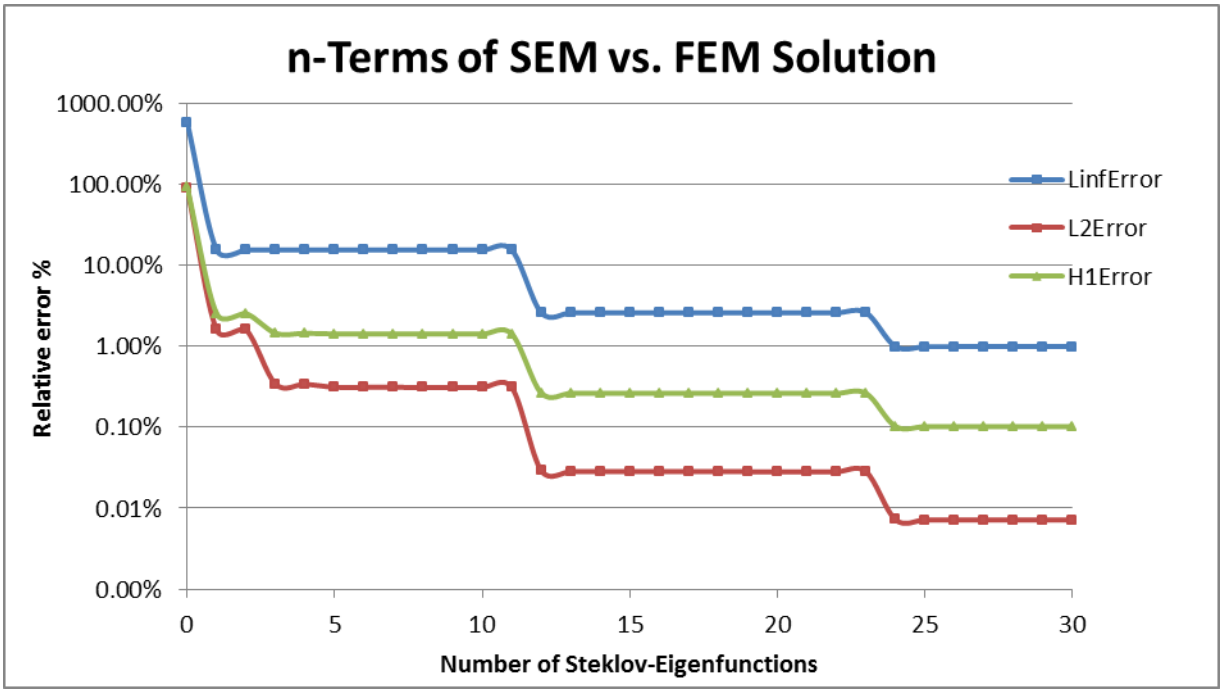


Figure 43: Relative errors for different norms comparing n-successive terms in the Steklov-expansion (x-axis) compared against solution produced by FEM (y-axis).

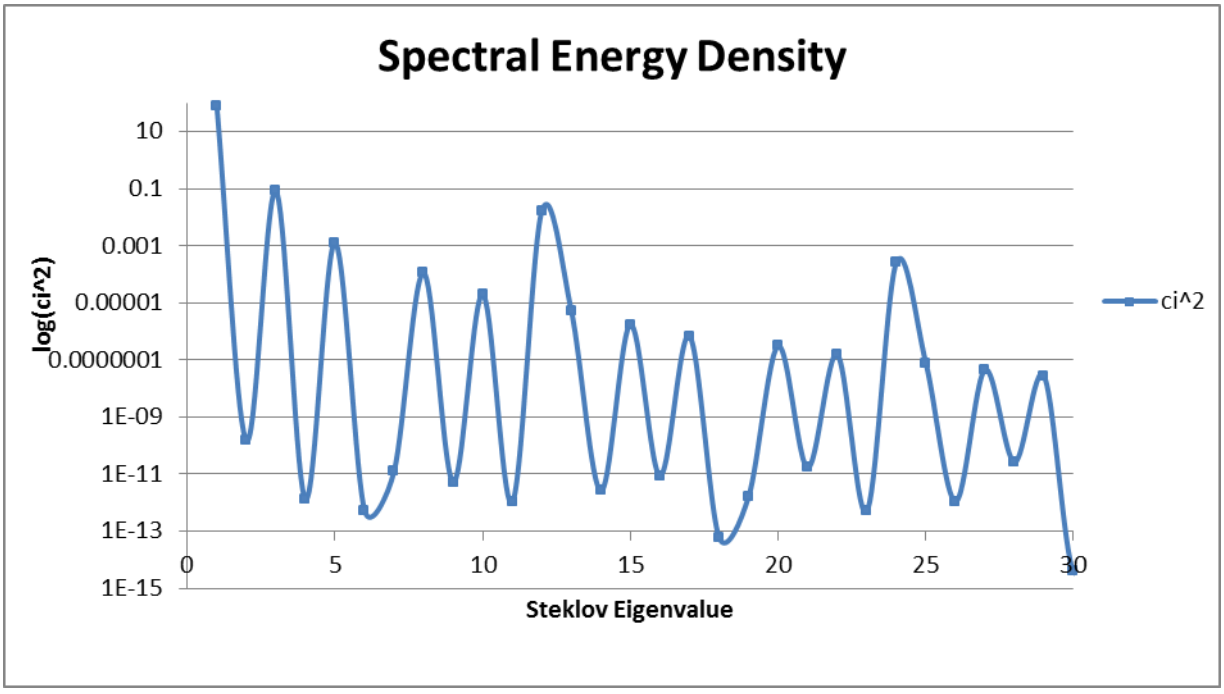


Figure 44: Spectral energy density given by $\langle s_n, g \rangle_{\Sigma}^2$.

	Steklov Eigenvalue	Relative L_∞-Error n SEM Terms Vs FEM Solution	Relative $L_2 - Error$ n SEM Terms Vs FEM Solution	Relative $H_1 - Error$ n SEM Terms Vs FEM Solution	n Coefficient in SEM Expansions
n	δ_n	$\frac{\|u_n^S - u_{FEM}^*\|_\infty}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^S - u_{FEM}^*\|_2}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^S - u_{FEM}^*\|_{1,2}}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\langle s_n, g\rangle_\Sigma$
0	-5.70E-15	5.87036	0.909452	0.927613	0
1	0.115833	0.156171	0.015969	0.025448	8.91375
2	0.310419	0.156166	0.015969	0.025448	1.24E-05
3	0.628316	0.155052	0.003409	0.014534	0.291046
4	0.942573	0.155052	0.003409	0.014534	-1.16E-06
5	1.25683	0.155052	0.003124	0.014265	0.036052
6	1.57099	0.155051	0.003124	0.014265	7.36E-07
7	1.57118	0.155051	0.003124	0.014265	3.57E-06
8	1.88574	0.155051	0.003107	0.014234	0.010681
9	2.20028	0.155051	0.003107	0.014234	2.31E-06
10	2.5156	0.155051	0.003104	0.014226	0.004502
11	2.83111	0.155051	0.003104	0.014226	-1.02E-06
12	3.1419	0.026052	0.00029	0.002649	0.128994
13	3.14667	0.026052	0.000285	0.002637	0.002305
14	3.46344	0.026052	0.000285	0.002637	1.62E-06
15	3.78087	0.026052	0.000283	0.002632	-0.00133
16	4.09998	0.026052	0.000283	0.002632	-2.99E-06
17	4.41843	0.026052	0.000283	0.00263	-0.00084
18	4.71291	0.026052	0.000283	0.00263	2.50E-07
19	4.73863	0.026052	0.000283	0.00263	-1.27E-06
20	5.06115	0.026052	0.000283	0.002628	0.000558
21	5.38563	0.026052	0.000283	0.002628	-4.17E-06
22	5.71145	0.026052	0.000282	0.002628	-0.00039
23	6.03549	0.026052	0.000282	0.002628	7.12E-07
24	6.2842	0.009927	7.19E-05	0.001024	-0.01612
25	6.36402	0.009927	7.18E-05	0.001023	0.000283
26	6.70131	0.009927	7.18E-05	0.001023	-1.06E-06
27	7.03337	0.009927	7.17E-05	0.001023	0.000213
28	7.36945	0.009927	7.17E-05	0.001023	-5.19E-06
29	7.70985	0.009927	7.16E-05	0.001022	-0.00016

Table 7: Steklov Eigenvalues for the inviscid flow between parallel plates with a contraction-expansion. Relative errors in different norms captured for n-successive terms in the Steklov-expansion compared against solution produced by FEM.

(v) Inviscid Fluid Flow around a Circle

This example demonstrates inviscid flow between two plates around a circle cut out in the center of the domain. The flow is assumed to be inviscid and incompressible planar flow. The geometry in this example is a rectangle with a circle cut from the center. The top and bottom wall plates and boundary of the circle are considered impermeable with a no-flux condition enforced on these walls. The remaining left and right walls continue to contain a velocity flux, $(v.n_1)(y) = \mu_1(y)$ and $(v.n_2)(y) = \mu_2(y)$, such that the net flux across the boundary $\int_{\partial\Omega}\mu_1\, dy - \int_{\partial\Omega}\mu_2\, dy$ is zero. The problem takes the form:

Let $v(x)$ be the unknown fluid velocity between the expanded plates in the planar region $\Omega = (-\mathrm{L}, 0) \times (-\mathrm{M}, \mathrm{M})/\mathrm{C}_0$ with $\mathrm{C}_0 \coloneqq \{(R\ cos\ 2\pi t, R\ sin\ 2\pi t) | t \in [0,1]\}$. R is restricted to be strictly inscribed in Ω. There is a velocity potential $\varphi(x)$, $v = \nabla\varphi$, that satisfies

$$\Delta\varphi(x) = 0, \quad x \in \Omega \tag{68}$$

subject to the Neumann boundary conditions

$$-\frac{\partial\varphi}{\partial x}(-L, y) = \mu_1(y) = \left(1 - \left(\frac{y}{\mathrm{M}}\right)^2\right), for\ \ y \in [-\mathrm{M}, \mathrm{M}]\,, \tag{69}$$

$$\frac{\partial\varphi}{\partial x}(L, y) = \mu_2(y) = \left(1 - \left(\frac{y}{\mathrm{M}}\right)^2\right), for\ \ y \in [-\mathrm{M}, \mathrm{M}],$$

$$\frac{\partial\varphi}{\partial y}(x, \mathrm{M}) = \frac{\partial\varphi}{\partial y}(x, -\mathrm{M}) = 0, \ for\ x \in [-L, 0], \tag{70}$$

$$\frac{\partial\varphi}{\partial v}(x, y) = 0\,, for\ (x, y) \in C_0. \tag{71}$$

The equivalent mixed variational formulation is to find $\varphi \in H^1(\Omega)$ and $\lambda \in \mathbb{R}$ such that

$$\int_\Omega \nabla\varphi \cdot \nabla\chi\ dx + \int_\Omega \varphi\beta\ dx + \int_\Omega \lambda\chi\ dx = \int_{\partial\Omega}\mu\chi\ d\sigma, \text{ for all } \chi \in H^1(\Omega), \beta \in \mathbb{R}. \tag{72}$$

```
macro Grad(u) [dx(u),dy(u)]
// Geometry - Solid Plate
real L = 2.0, M=2.0;  // L - value
border Left (t=0,1) {x=-L;y=M*(1.0-2.0*t);label=1;}  // Left barrier
border Top  (t=0,1) {x=L*(1.0-2.0*t);y=M;label=2;}   // Top wall
border Right(t=0,1) {x=L;y=M*(2.0*t-1.0);label=3;}   // Right barrier
border Bottom(t=0,1){x=L*(2.0*t-1.0);y=-M;label=2;}  // Bottom wall
border C (t=0,1) {x=0.25*cos(2.0*pi*t);y=0.25*sin(2.0*pi*t); label=4;}        // Circular obstacle

// Construct a mesh
mesh Th = buildmesh(Left(49) + Top(49) + Right(49) + Bottom(49) + C(-49) );
// Define inlet/outlet
func g1 = 1-(y/M)^2;
func g2 = (1-(y/(5*M))^2)/5;
// Finite element and functions
fespace Vh(Th,P2);
Vh uhFEM,uhSEM,vh;                          // Holds the final uhFEM
int eigCount=33, numEigs=33, count=0; // total number of eigenvalues
real[int] ev(eigCount);                     // Eigenvalues
Vh [int]   eV(eigCount),eVold(eigCount),pSum(eigCount);  // Eigenvectors
// Steklov - Eigenvalue problem in variational form
real L2Errorsq = 1.0E8, adaptErr = 1.0e-2, shift = 0;
varf vA(u,v) = int1d(Th, 1,3)(-shift* u*v)+int2d(Th)(dx(u)*dx(v)+dy(u)*dy(v));
varf vB(u,v) = int1d(Th, 1,3)( u*v);
while(L2Errorsq > 5.0E-2)
{
  if (count > 0 && count < 5) // Adaptation step
  {
     Vh fAdapt = 0;
     for(int i=1;i<numEigs;i++)
        fAdapt = fAdapt +  abs(eV[i]);
     Th = adaptmesh(Th,fAdapt,err=adaptErr);
     adaptErr/=2.0;
  } else if (count > 0) Th = splitmesh(Th,2);
  // Get resulting matrices Ax=lBx
  matrix A = vA(Vh,Vh,solver = sparsesolver);
  matrix B = vB(Vh,Vh);
```

```
  // Solve Ax=lBx
  numEigs = EigenValue(A,B,sym=true,sigma=0,value=ev,vector=eV);
  numEigs = min(eigCount,numEigs);
  L2Errorsq = 0.0;
  for(int i=0;i<numEigs;i++)
  {
     Vh ssDiff = abs(abs(eV[i])-abs(eVold[i]));
     L2Errorsq = max(L2Errorsq,int2d(Th)(ssDiff*ssDiff));
     eVold[i] = eV[i];
  }  count++;
}
// Construct solution
real[int] c(numEigs);
for(int i=0;i<numEigs;i++)
{
  c[i] =(int1d(Th, 1)(g1*eV[i]) + int1d(Th, 3)(g2*eV[i]))/ ev[i];
  uhSEM = uhSEM + c[i] * eV[i];
  pSum[i] = uhSEM;
}
// Solve mixed DN FEM problem
varf va(uh,vh) = int2d(Th)(Grad(uh)'*Grad(vh));
varf vb(uh,vh) = int2d(Th)(vh);
varf vL(uh,vh) = -int1d(Th,1)(g1*vh) +  int1d(Th,3)(g2*vh);  // right wall
// Capture # of degrees of freedom to construct mixed matrix
int n1 = Vh.ndof+1;
// Construct mixed matrix to solve mixed FEM problem
//    A*uh + B*l = b
//    B*uh=0
matrix   A = va(Vh,Vh);   // Stiffness Block matrix A
real[int] b = vL(0,Vh);    // Solution vector
real[int] B = vb(0,Vh);    // Constraint block matrix
real[int] bb(n1),   xx(n1), l(1);   // Modified sol vector for mixed problem, sol vector, lagrange multiplier
matrix   AA = [[A,B],[B',0]];  // Mixed block matrix
set(AA,solver=sparsesolver); // Set solver type on matrix
bb = [b,0];      // Construct right hand side vector
xx = (AA^-1)*bb; // Get solution vector
[uhFEM[],l] = xx;   // Part of vector has solution, other has Lagrange multiplier
// Plot solutions
plot(uhSEM,value=true,fill=true,ColorScheme=2);
plot(uhFEM,value=true,fill=true,ColorScheme=2);
```

Code Snippet 14: Code to solve Neumann BVP with FEM and adapted SEM.

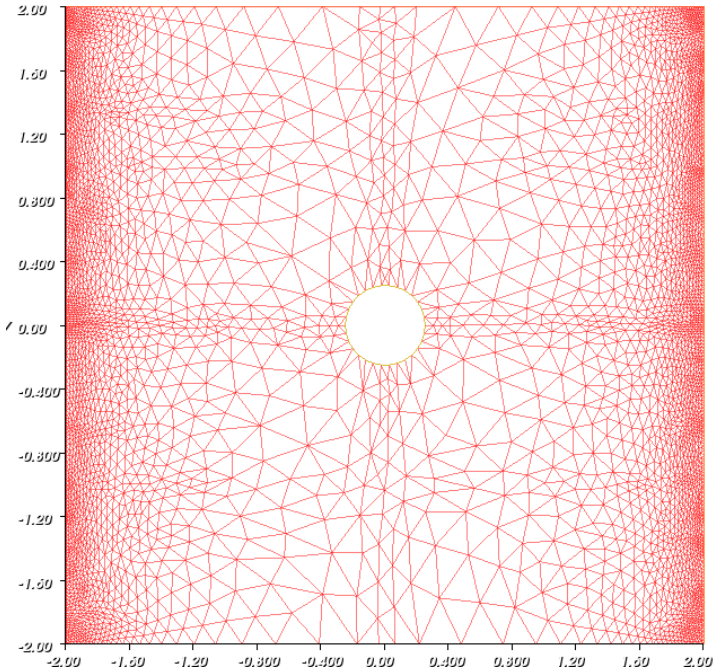
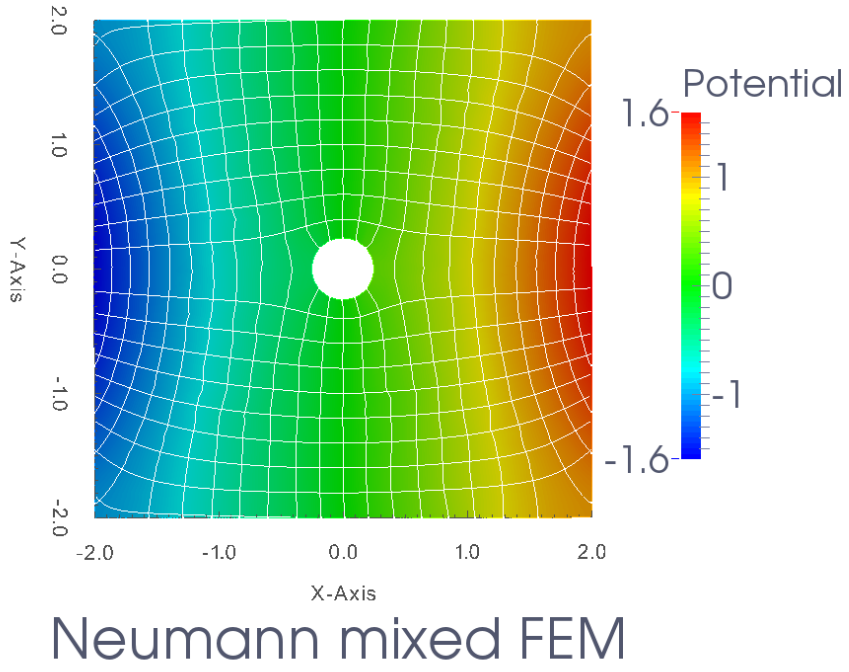


Figure 45: Adapted mesh for SEM method



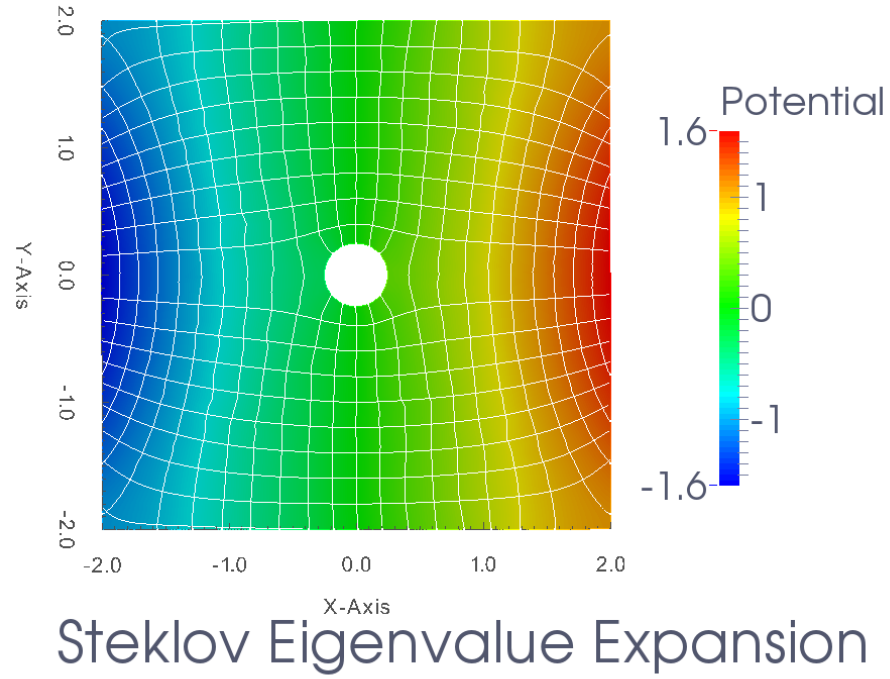


Figure 46: Solutions from both SEM & FEM procedures. Horizontal particle streamlines and vertical contour lines are overlayed.

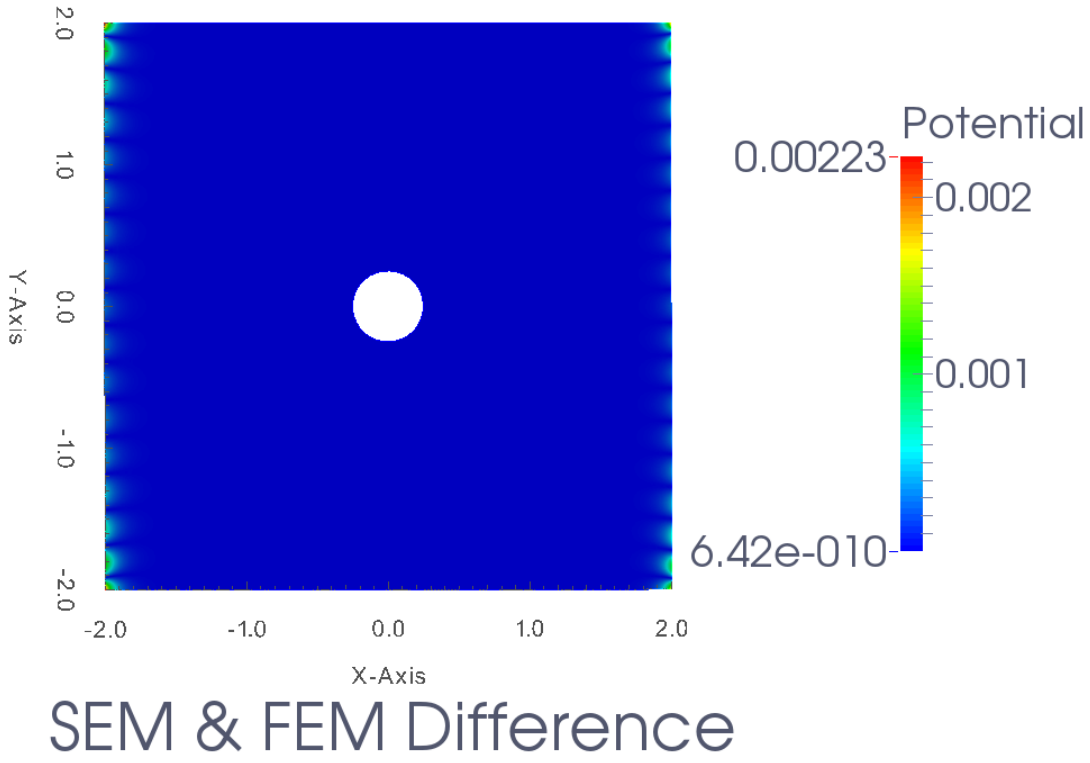


Figure 47: Absolute difference of SEM & FEM solutions. Some difference is revealed on the boundary.

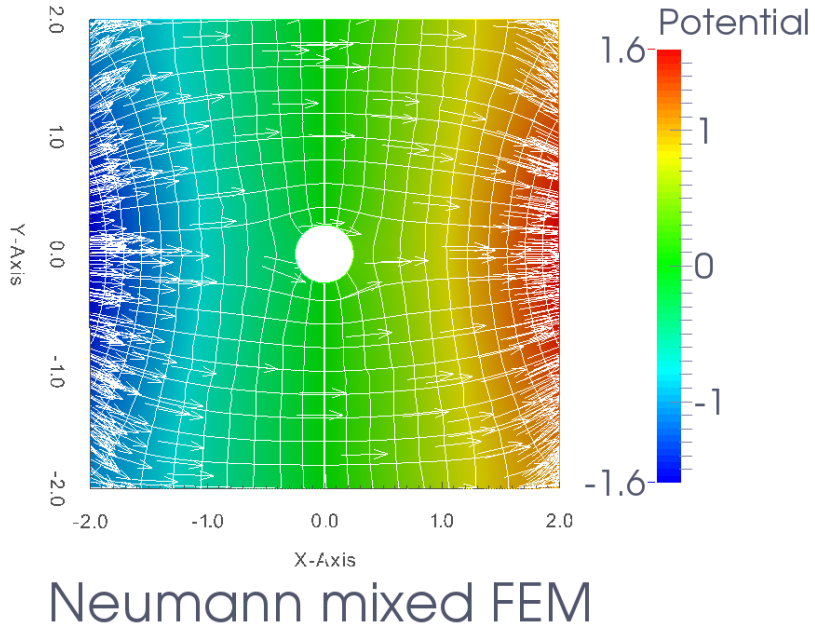


Figure 48: Vector field of potential showing flow of fluid across a circle.

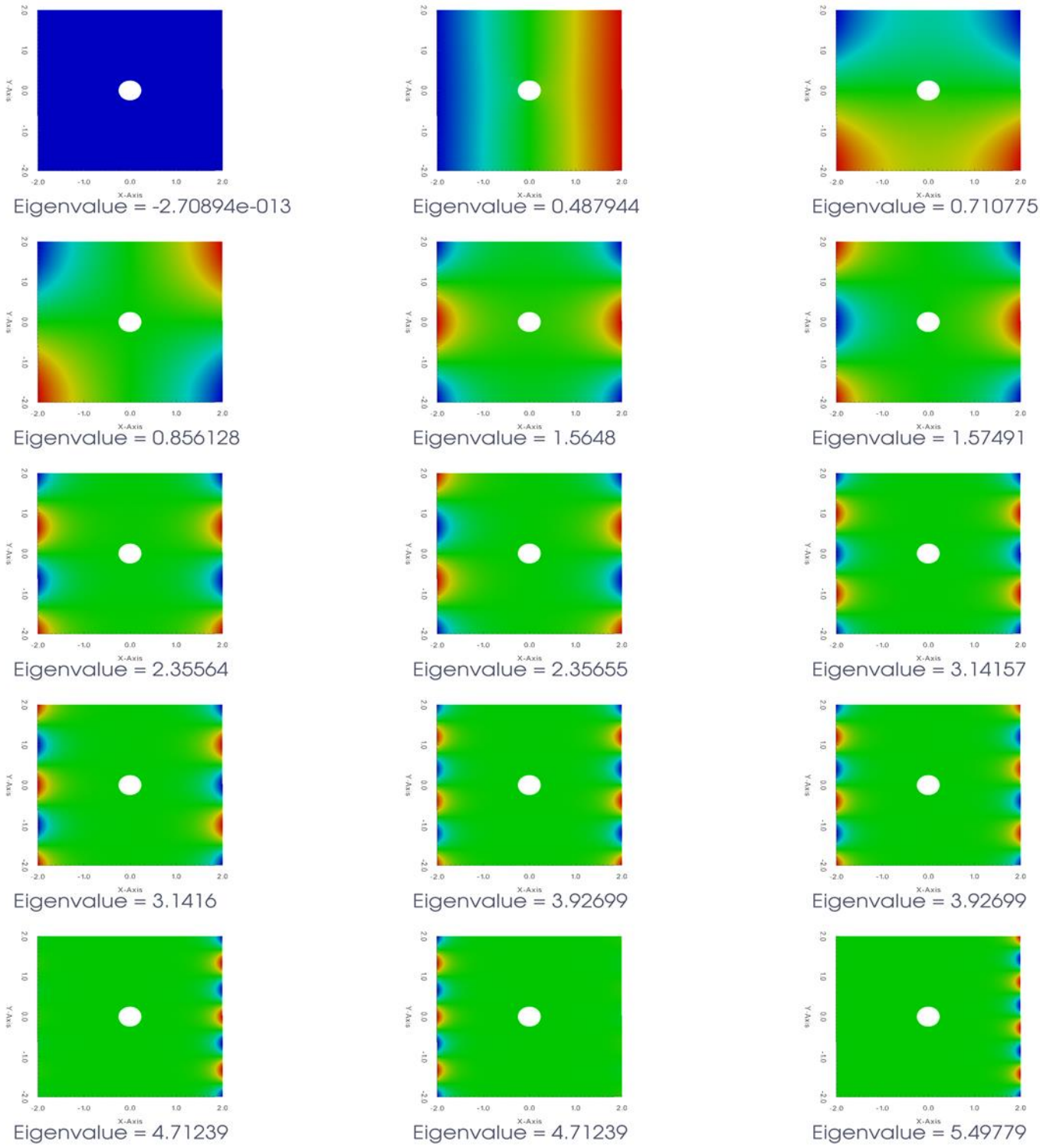


Figure 49: First 15 ordered Steklov-eigenfunctions (left to right) of the Inviscid flow around a Circle problem.

Figure 50: Next 15 ordered Steklov-eigenfunctions (left to right) of the inviscid flow around a circle problem. Notice the localization near the boundary of higher order terms.

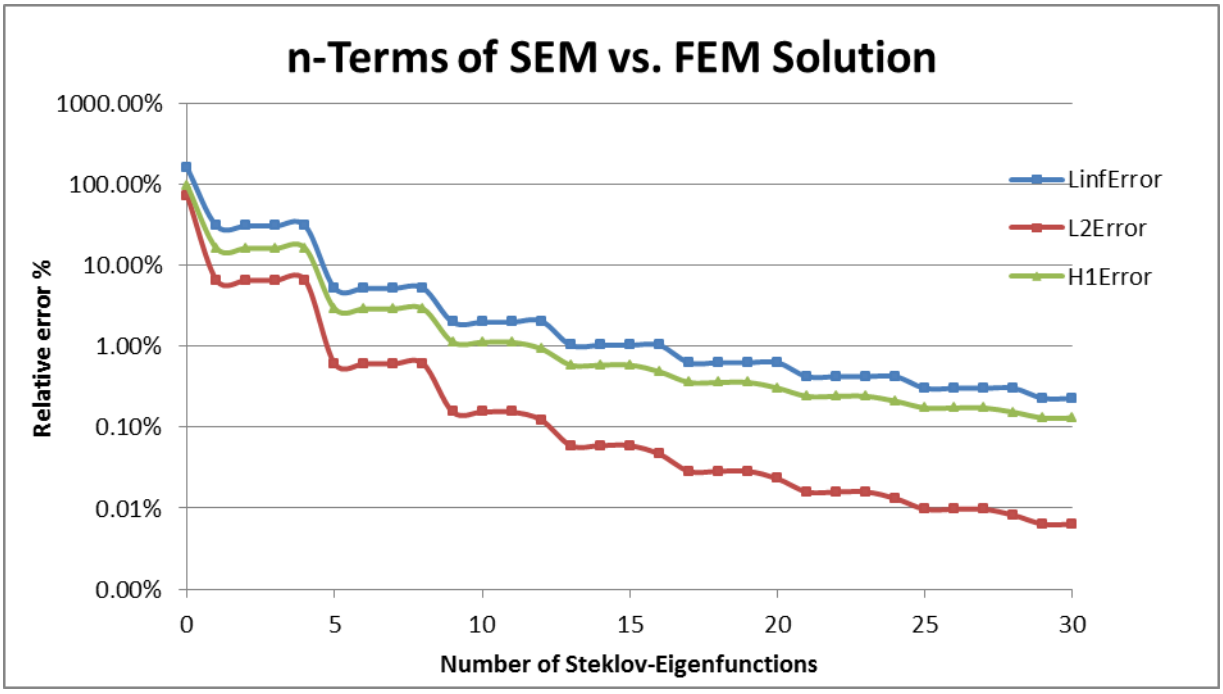


Figure 51: Relative errors for different norms comparing n-successive terms in the Steklov-expansion (x-axis) compared against solution produced by FEM (y-axis).

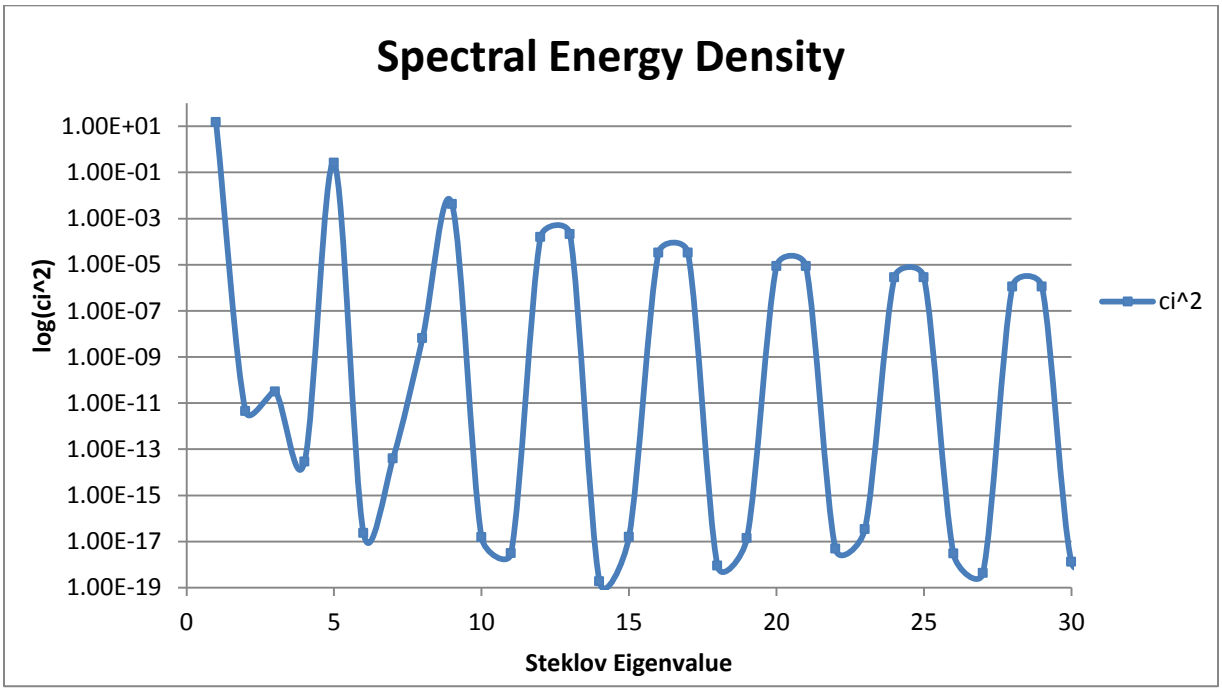


Figure 52: Spectral energy density given by $\langle s_n, g \rangle_{\Sigma}^2$.

	Steklov Eigenvalue	Relative L_∞-Error n SEM Terms Vs FEM Solution	Relative $L_2 - Error$ n SEM Terms Vs FEM Solution	Relative $H_1 - Error$ n SEM Terms Vs FEM Solution	n Coefficient in SEM Expansions
n	δ_n	$\frac{\|u_n^S - u_{FEM}^*\|_\infty}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^S - u_{FEM}^*\|_2}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\frac{\|u_n^S - u_{FEM}^*\|_{1,2}}{\|u_{FEM}^*\|_{\partial\Omega}}$	$\langle s_n, g\rangle_\Sigma$
0	-2.71E-13	1.60586	0.735374	0.971806	0
1	0.487944	0.307714	0.065563	0.162729	3.8714
2	0.710775	0.307713	0.065563	0.162729	2.12E-06
3	0.856128	0.30771	0.065563	0.162729	-5.63E-06
4	1.5648	0.30771	0.065563	0.162729	1.69E-07
5	1.57491	0.052189	0.006097	0.029102	0.509559
6	2.35564	0.052189	0.006097	0.029102	-4.80E-09
7	2.35655	0.052189	0.006097	0.029102	-2.00E-07
8	3.14157	0.052229	0.006097	0.029102	8.06E-05
9	3.1416	0.01988	0.001543	0.011247	0.064619
10	3.92699	0.01988	0.001543	0.011247	-3.90E-09
11	3.92699	0.01988	0.001543	0.011247	-1.75E-09
12	4.71239	0.020519	0.00123	0.009333	0.012501
13	4.71239	0.010326	0.000594	0.005872	-0.01445
14	5.49779	0.010326	0.000594	0.005872	-4.29E-10
15	5.49779	0.010326	0.000594	0.005872	3.99E-09
16	6.28319	0.010326	0.000466	0.004866	-0.0057
17	6.28319	0.006294	0.000286	0.003588	0.005702
18	7.06859	0.006294	0.000286	0.003588	9.48E-10
19	7.06859	0.006294	0.000286	0.003588	3.72E-09
20	7.85399	0.006294	0.000232	0.003058	0.002919
21	7.85399	0.00423	0.000159	0.002414	-0.00292
22	8.63939	0.00423	0.000159	0.002414	2.19E-09
23	8.6394	0.00423	0.000159	0.002414	5.87E-09
24	9.42479	0.00423	0.000132	0.002102	0.001689
25	9.4248	0.003036	9.70E-05	0.001734	0.001689
26	10.2102	0.003036	9.70E-05	0.001734	1.73E-09
27	10.2102	0.003036	9.70E-05	0.001734	-6.57E-10
28	10.9956	0.003036	8.20E-05	0.001534	-0.00106
29	10.9956	0.002283	6.35E-05	0.001305	-0.00106

Table 8: Steklov eigenvalues for the inviscid flow around a circle. Relative errors in different norms captured for n-successive terms in the Steklov-expansion compared against solution produced by FEM.

9. Summary & Conclusions

For two-dimensional regions, SEM is a viable methodology for calculating mixed DN BVPs and special types of harmonic vector field BVPs. Analytical results for the eigenvalues for the cases with rectangular geometry were closely approximated by results using SEM. Results using the direct FEM approach also agree with SEM eigenfunction expansion constructions. The qualitative behavior observed in the electrostatic, contraction-expansion between parallel plates and flow around a circle align with expected results predicted in fluid mechanics and electrostatics. For the harmonic vector field cases, the fields were successfully constructed from the gradient of Steklov-eigenfunction expansions.

The number of eigenfunction expansion terms is small and depends highly on the oscillatory nature of the boundary condition. The higher the oscillation, the higher the number of terms involved in the eigenvalue expansion. Even though a higher order eigenfunction may be required to completely capture the boundary, the function on the interior of the domain is reproduced in general with 10 to 15 terms. The larger eigenfunctions appear to be localized near the boundary while lower ordered terms cover the general domain. This provides some evidence that boundary layer phenomenon is properly captured by Steklov-eigenvalue expansion techniques.

Each example calculated in section 7 show good L2-converge while L-max and H1-convergence is less favorable. The current calculation of the H1-norm might need to be modified as it involves taking derivatives which is prone to numerical error. L2-convergence results indicate that Steklov-eigenfunctions appear to work in groups. Once a certain level of convergence is achieved, a notable drop is noticed after an additional number of terms are included (depending on the group size). This is indicated in the plot below for reference.

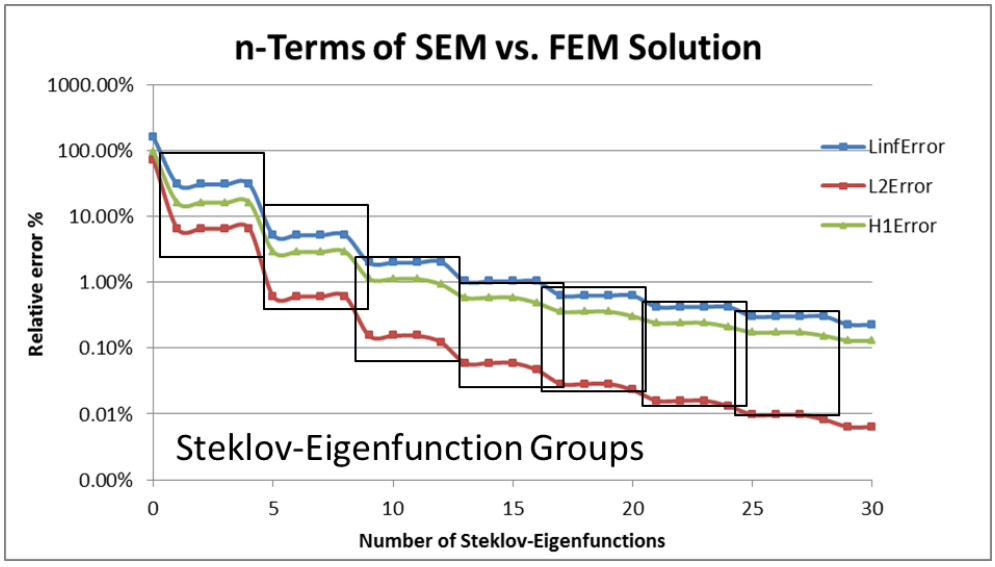


Figure 53: L2-convergence rates for the three harmonic BVP problems.

The spectral density plots reveal how the Steklov-eigenfunction terms respond to boundary conditions, how they are involved in the construction of the solution and how they individually contribute to the coefficient decay-rates. The three harmonic vector field examples (parallel plates, contraction expansion, and flow around a circle) share similar boundary conditions. The plot below reveals a relationship between spectral energy density and the problem geometry. The second group of eigenfunctions appears to reveal the change in geometry experienced in the flow around a circle and contraction expansion examples. The magnitude of the spectrum also is impacted by the region size. The oscillations and grouping pattern seen below is an indication of the influence of the boundary.

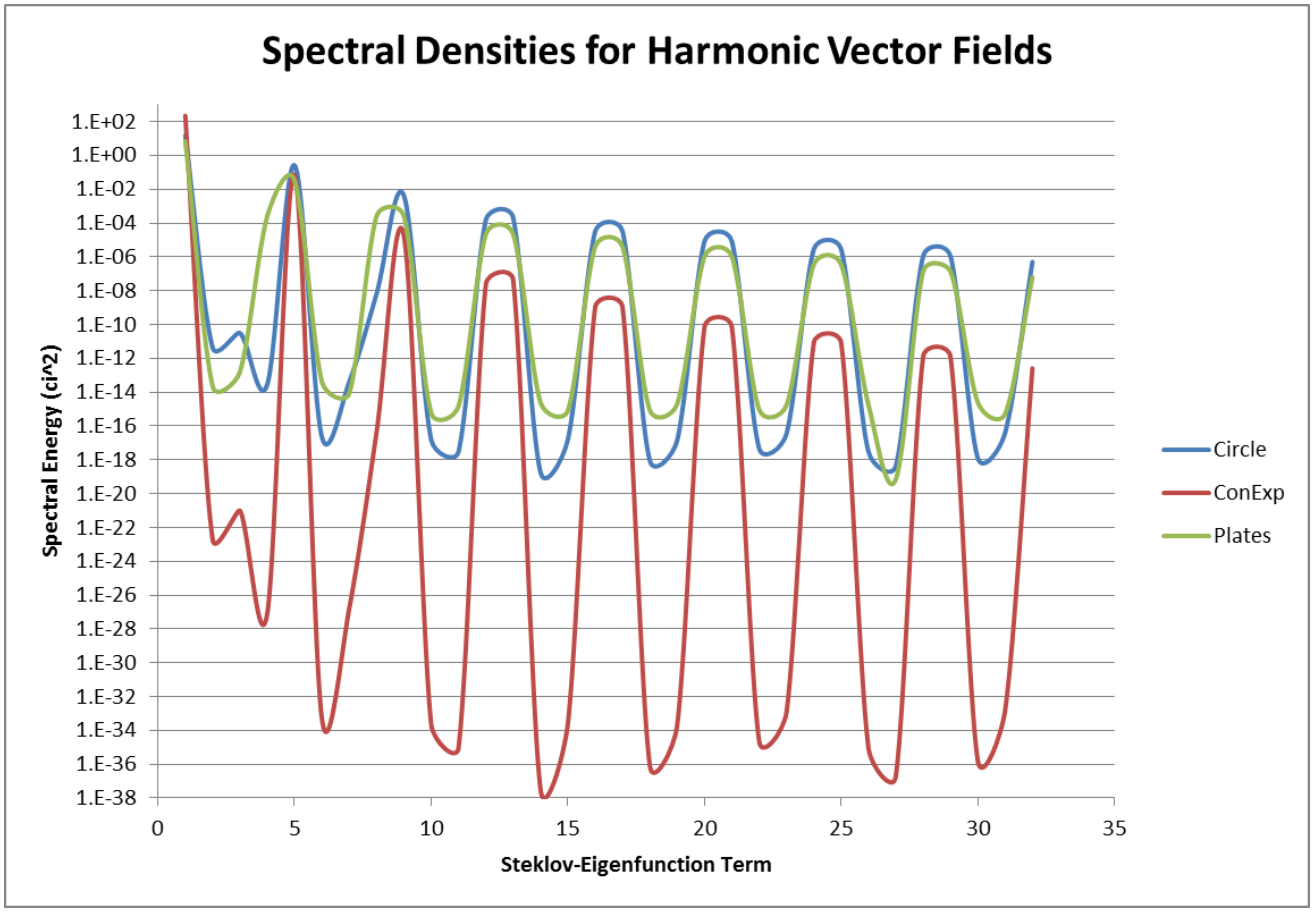


Figure 54: Spectral densities for the three harmonic vector field problems.

The only major challenge observed in using SEM versus traditional FEM and mixed FEM approaches is the difficulty in capturing the localization of higher order Steklov-eigenfunctions. A single mesh that is able to capture an entire range of Steklov-eigenfunctions and converge within bounds similar to FEM requires further investigation. The localization of higher order Steklov-eigenfunction causes sharp gradients near the boundaries that can easily be missed if the mesh is not properly defined. Using a uniform mesh becomes problematic as the number of finite elements becomes large. An adaptive technique is the correct approach to take. This project did not include a proper adaptation mythology. A posterior error estimate needs to be developed to properly create an adaptive algorithm. This offers another area to further investigate.

SEM also has some promising implications. Once the eigenfunction and eigenvalue spectrum is calculated, the intermediate results can be used to quickly reconstruct solutions. For problems where the spectrum can be pre-calculated and applied to different boundary conditions, this approach is preferable to recalculating FEM several times as the recalculation using SEM only requires evaluating boundary integrals. This approach is a competitor of the traditional boundary element method (BEM). BEM has the advantage of boundary integrals but suffers from integrating a potential which contains a singularity. The BEM method also results in a full matrix while the SEM method results in a nice sparse matrix. For time-dependent problems, the SEM approach may possess similar advantages. If the spectrum only has to be calculated once, then there could be a major advantage of using SEM for problems that involve time. Overall, SEM is a promising alternative for many types of problems.

10. Further Work

(i) Extending to Axisymmetric

In addition to solving planar flows, it is straightforward to extend SEM to solve harmonic bvps with axisymmetric cylindrical geometry. This example will demonstrate how to calculate the Steklov-eigenvalues and eigenfunctions in cylindrical coordinates assuming the Steklov-eigenfunctions do not vary in the tangential direction. The geometry is assumed to be a cylinder or radius R and length L. The outer wall of the cylinder is assumed to be impermeable while the left and right wall corresponding to the top and bottom are assumed to contain a flow with prescribed flux such that net flux across the boundaries is zero. The geometry used to form the mesh is identical to the geometry used in the parallel plates and solid plate heat conduction problems. The variational form is altered to account for axisymmetric cylindrical coordinates. The Steklov-eigenvalue problem takes the classical form:

Let $v(x)$ be the unknown fluid velocity in the cross-sectional region $\Omega = (0, \mathrm{R}) \times (0, \mathrm{L})$ of a cylinder. There is a velocity potential $\varphi(x)$, $v = \nabla\varphi$, that satisfies

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \varphi}{\partial r}\right) + \frac{\partial \varphi}{\partial r} = 0, \quad (r, z) \in \Omega \tag{73}$$

subject to the Neumann boundary conditions

$$\begin{aligned} -\tfrac{\partial \varphi}{\partial z}(r, 0) &= \delta\varphi(r, 0), \\ \frac{\partial \varphi}{\partial r}(r, L) &= \delta\varphi(r, L), for\ r \in [0, R], \\ \tfrac{\partial \varphi}{\partial r}(R, z) &= 0, \ for\ z \in [0, L]. \end{aligned} \tag{74}$$

The equivalent mixed variational formulation is to find $\varphi \in H^1(\Omega)$ and $\delta \in \mathbb{R}$ such that

$$\int_\Omega \mathrm{r}\, \nabla\varphi \cdot \nabla\chi \ d(r, z) = \delta \int_\Sigma r\varphi\chi \, d\sigma, \text{ for all } \chi \in H^1(\Omega). \tag{77}$$

Σ here is the left and right walls.

The FreeFEM++ code to solve the problem is shown below. No mesh adaptation has been used. A uniform dense grid is constructed instead.

```
real      L   = 4.0;
real      R = 1.0;
border Left(t=0, R)       {x=-0;  y= R-t;   label=1;}    // Left wall
border Top(t=0, L)        {x= L -t ; y= R;   label=2;}  // Top  wall
border Right(t=0, R)     {x=L;        y=t;     label=3;}   // Right wall
border Bottom(t=-0, L){x=t;     y=- 0; label=4;}  // Bottom wall

// Construct a mesh
mesh Th = buildmesh(Left(50) + Top(5*L) + Right(50) + Bottom(50));
fespace Vh (Th,P1);  // Create a P1-Lagrange FEM space
Vh uh, vh;                  // Instantiate instances of finite element space

// Create variational form for Steklov-Eigenvalue problem
// x is z, y is r
varf va(uh, vh) = int2d(Th)( y*( dx(uh)*dx(vh)+dy(uh)*dy(vh)));
varf vb(uh, vh) = int1d(Th,1,3)(y* uh * vh);

// Construct matrices to solve eigenvalue problem
//    A*x=l*B*x
matrix A = va(Vh, Vh ,solver = sparsesolver);     // Matrix A on left hand side
matrix B = vb(Vh, Vh);                                    // Matrix B on right hand side

int eigCount = 33;           // Get first 6 Eigenvalues
real[int] ev(eigCount);   // Holds Eigenfunctions
Vh[int] eV(eigCount);         // Holds Eigenvalues

// Solve Ax=lBx
int numEigs = EigenValue(A,B,sym=true,sigma=0,value=ev,vector=eV);
numEigs = min(eigCount,numEigs);

for(int i=0;i<numEigs;i++)  // Plot the spectrum
    plot(eV[i],fill=true,value=true,cmm= ev[i]);
```

Code Snippet 15: Code that demonstrates solving an axisymmetric Steklov-eigenvalue problem.

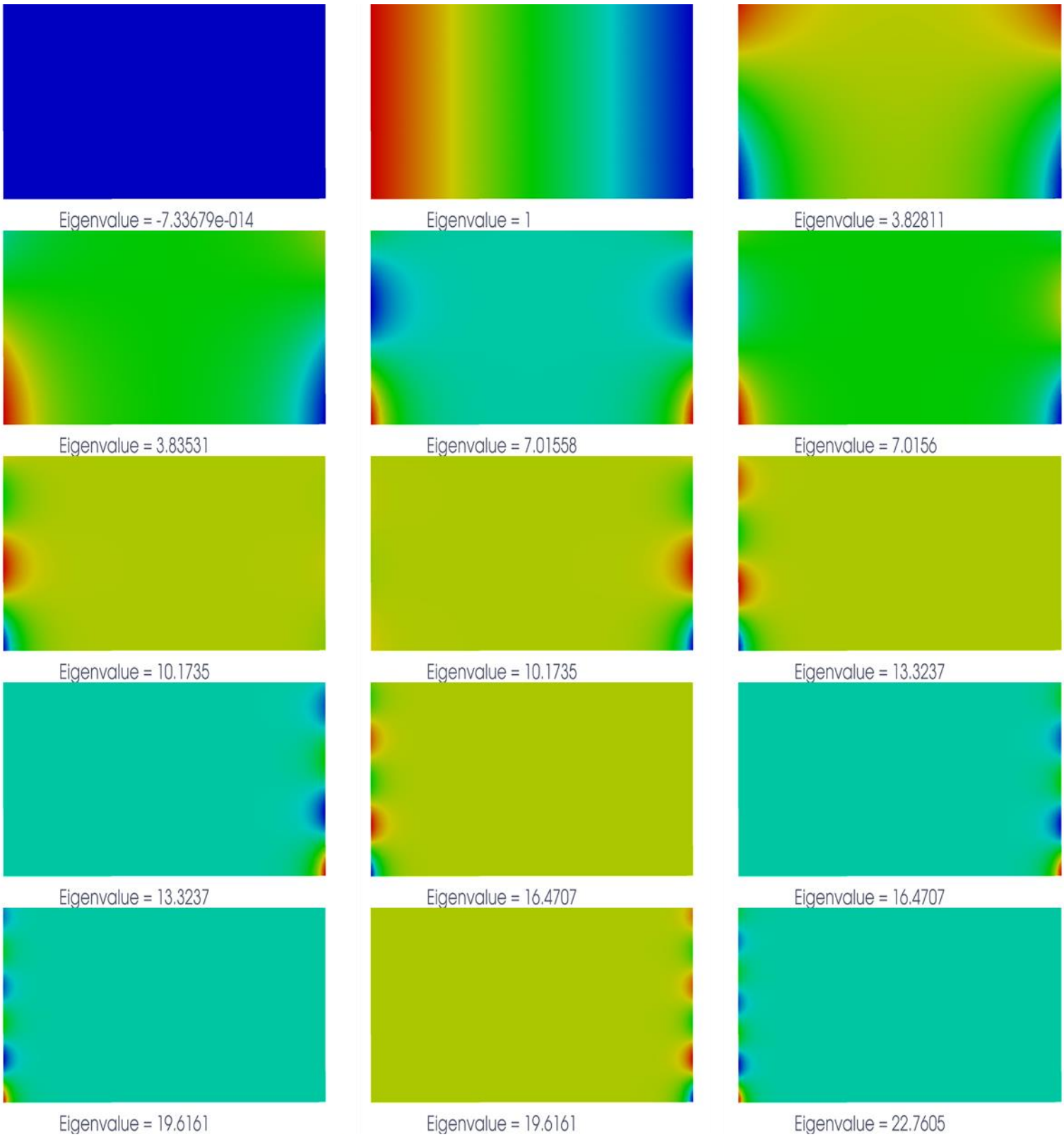


Figure 55: First 15 ordered axisymmetric Steklov-eigenfunctions (left to right).

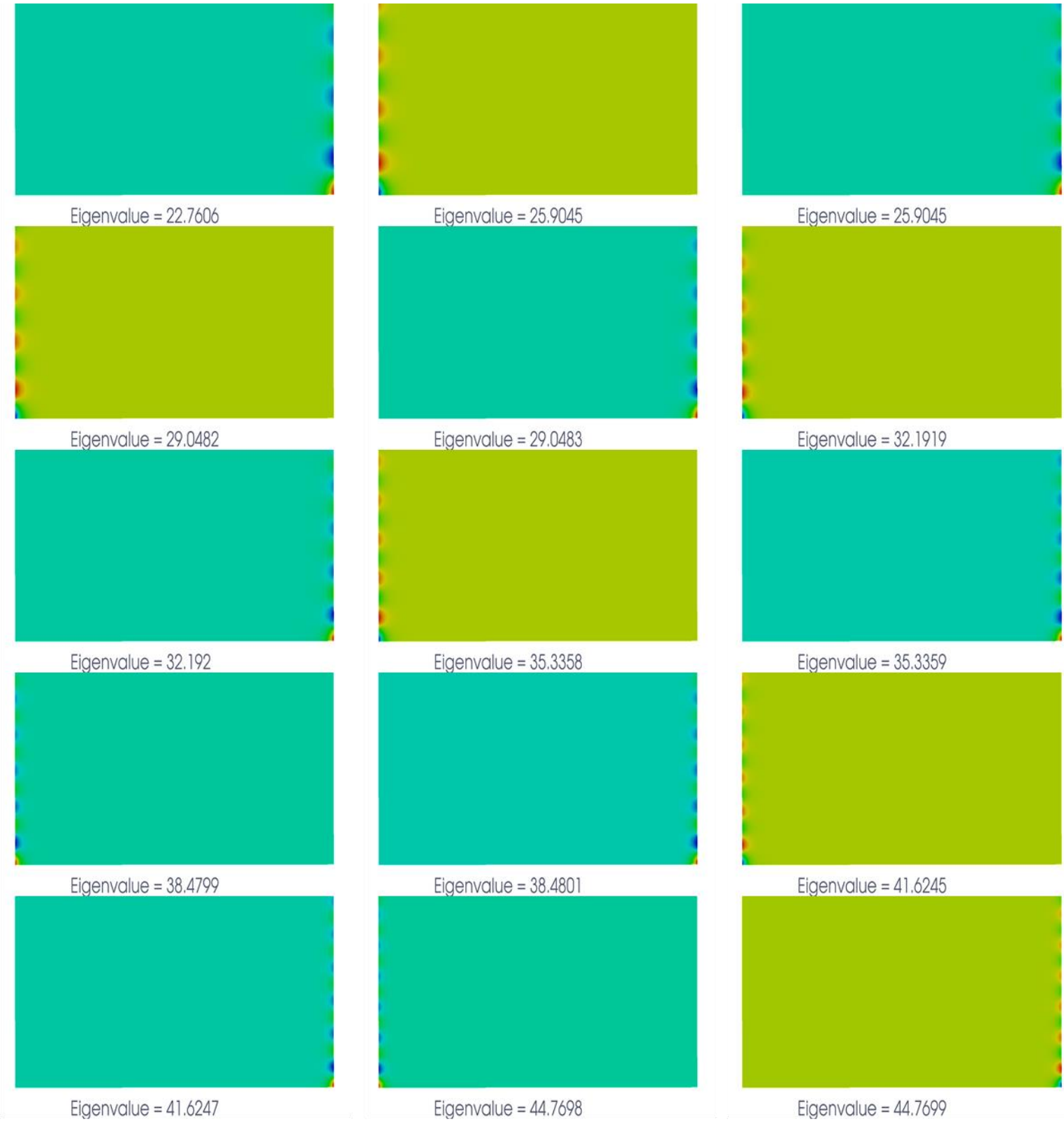


Figure 56: Next 15 ordered axisymmetric Steklov-eigenfunctions (left to right) .

(ii) Extending to 3D

In addition to solving axisymmetric flows, it is also straightforward to extend SEM to solve harmonic bvps in a 3D cylinder. The parallel plate's formulation can easily be generalized to 3D. Here, flow is allowed across cylinder wall. This is to simplify demonstration. The Steklov-eigenvalue problem takes the classical form:

Let $v(x)$ be the unknown fluid velocity in a cylinder $\Omega = (0, \mathrm{R}) \times (0,2\pi) \times (0, \mathrm{L})$. There is a velocity potential $\varphi(x)$, $v = \nabla\varphi$, that satisfies

$$\Delta\varphi(x) = 0, \quad x \in \Omega \tag{78}$$

subject to the Neumann boundary conditions

$$\frac{\partial\varphi}{\partial n} = \delta\varphi \ \text{ on } \partial\Omega. \tag{79}$$

The equivalent mixed variational formulation is to find $\varphi \in H^1(\Omega)$ and $\delta \in \mathbb{R}$ such that

$$\int_\Omega \nabla\varphi \cdot \nabla\chi \ dx = \delta \int_{\partial\Omega} \varphi\chi \, d\sigma, \text{ for all } \chi \in H^1(\Omega). \tag{80}$$

It is a little bit more involved to create a 3D mesh in FreeFEM++. The cylinder mesh will be created using external library calls to software called "medit" and "tetgen." "medit" is used for generating the perimeter, similar to the border command. "tetgen" is a program used to tetrahedralize the cylinder. All integrals are 3D now. The FreeFEM++ code to solve the problem is shown below. No mesh adaptation has been used.

```
load "msh3"
load "tetgen"
load "medit"
int nx=10;  // Number of points along axis
int nth=50; // Number of points along circumference
real xmin=1.,xmax=3.;
border cc(t=0,2*pi){x=cos(t);y=sin(t);label=1;}  // Create a circle border
mesh Thcircle = buildmesh(cc(nth));  // Build a 2D mesh from the circle
mesh Thsquare=square(nx,nth,[xmin+x*(xmax-xmin),2*pi*y]);  // Build a rectangle mesh next
// Parameterization construct a 3D mesh from 2 2D meshes
func f1 = x; func f2 = cos(y); func f3 = sin(y);  // Cylindrical coordinates
mesh3 Thsurf1=movemesh23(Thsquare,transfo=[f1,f2,f3],orientation=-1); // Cylinder surface
mesh3 Thsurf2=movemesh23(Thcircle,transfo=[xmin,x,y],orientation=-1); // Top
mesh3 Thsurf3=movemesh23(Thcircle,transfo=[xmax,x,y],orientation=1);  // Bottom
mesh3 Thsurf=Thsurf1+Thsurf2+Thsurf3;  // Put the meshes together
real voltet= ( ( (2*pi)/20 )^3 )/6.; real[int] domaine = [1.5,0.,0.,1,voltet]; // Tetgen params
mesh3 Th=tetg(Thsurf,switch="pqaaAAYYQ",nbofregions=1,regionlist=domaine); // Mesh it!

fespace Vh(Th,P2);  // P2 finite element space
// Steklov-variational problem
varf vA(u,v) = int3d(Th)(dx(u)*dx(v)+dy(u)*dy(v)+dz(u)*dz(v));
varf vB(u,v) = int2d(Th)(u*v);
// Convert variational forms to matrices
matrix A=vA(Vh,Vh,solver=sparsesolver);
matrix B=vB(Vh,Vh);

// Calculate first 15 eigenvalues/eigenfunctions
int eigCount = 15;
real[int] ev(eigCount); Vh[int]   eV(eigCount);

// Solve the eigenvalue problem
int numEigs = EigenValue(A,B,sym=true,sigma=0,value=ev,vector=eV);
numEigs = min(eigCount,numEigs);

// Plotting parameters to slice a plane in picture
real[int] CutPlaneOriginValue = [0.0,0.0,0.0];
real[int] CutPlaneNormalValue = [0.0,0.0,1.0];
for(int i=0;i<numEigs;i++)
  plot(eV[i],cmm=ev[i], dim=3,value=true,boundary=0,CutPlane=0,
       CutPlaneOrigin=CutPlaneOriginValue,CutPlaneNormal=CutPlaneNormalValue,ColorScheme=2);
```

Code Snippet 16: Code that demonstrates solving a 3D Steklov-eigenvalue problem on a cylinder.

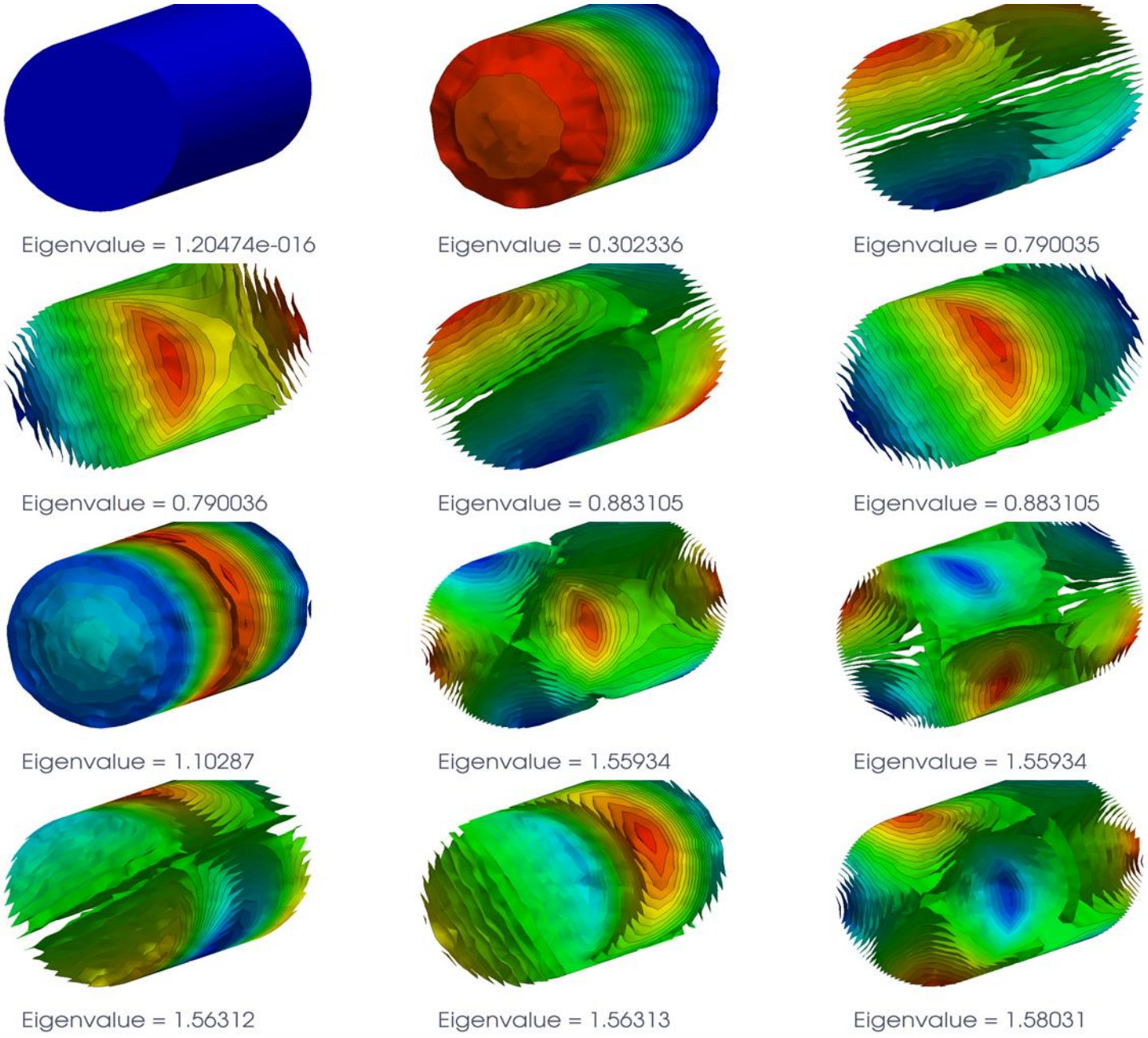


Figure 57: First 15 ordered Steklov-eigenfunctions (left to right) of a cylinder.

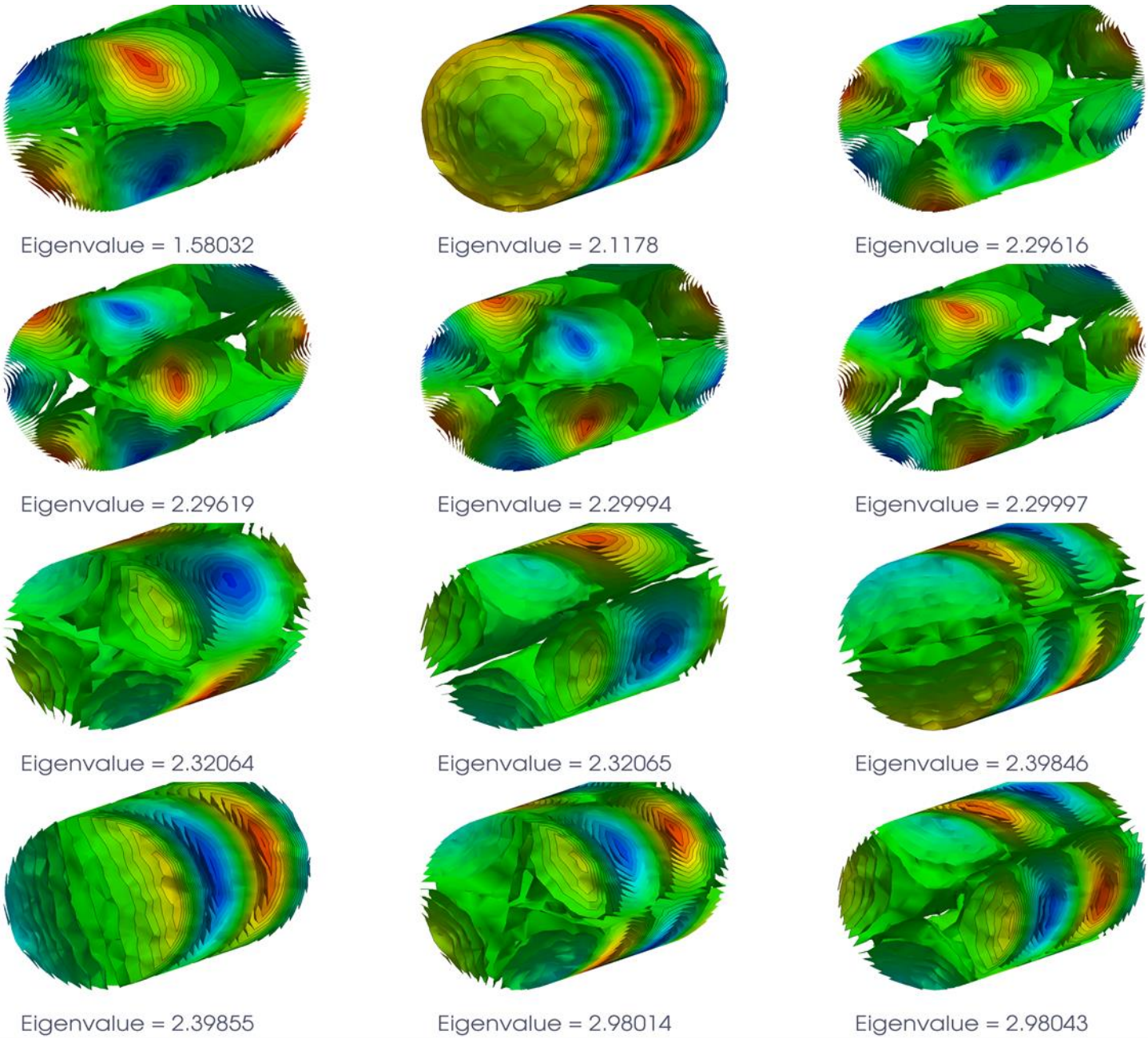


Figure 58: Next 15 ordered Steklov-eigenfunctions (left to right) of a cylinder.

11. References

[1] F. Hecht, A. Le Hyaric, K. Ohtsuka and O. Pironneau, "FreeFEM++ v 3.29," 17 3 2014. [Online]. Available: http://www.freemfem.org/ff++/ftp/freefem++doc.pdf. [Accessed 7 4 2014].

[2] G. Auchmuty and J. C. Alexander, "L2-well-posedness of planar div-curl systems," *Arch Rat Mech & Anal, 160,* pp. 91-134, 2001.

[3] G. Auchmuty, "Finite energy solutions of self-adjoint elliptic mixed boundray value problems," *Mathematical Methods in Applied Sciences,* vol. 33, no. 12, pp. 1446-1462, 2010.

[4] G. Auchmuty, "Steklov Eigenproblems and the Representation of Solutions of Elliptic Boundary Value Problems," *Numerical Functional Analysis and Optimization,* vol. 25, pp. 321-348, 2004.

[5] H. Atouch, B. Giuseppe and G. Michaille, Variational Analysis in Sobolev and BV Spaces: Applications to PDEs and Optimization (MPS-Siam Series on Optimization 6), SIAM, 2005.

[6] R. Glowinski, Numerical Methods for Nonlinear Variational Problems, Berlin-Heidelberg-New York-Tokyo: Springer-Verlag, 1984.

[7] F. Hecht, "New development in freefem++," *Numer. Math.,* vol. 20, no. 3-4, pp. 251-265, 2012.