École Polytechnique Fédérale de Lausanne

Eigenvalues of the Maxwell Cavity

MATH-468: Numerics for Fluids, Structures and Electromagnetics

---

Giulia Mescolini, Thomas Rimbot

14th January, 2022

Chapter 1: Analytical Setting

1.1 The framework

The object of our study is the *Maxwell cavity eigenvalue problem*, a fundamental benchmark case in Numerics and Mathematical Analysis applied to Electromagnetics.
The problem is set in a domain $\Omega \subset \mathbb{R}^2$ **regular** and **simply connected** and it arises from the so-called *magnetostatic problem*, where H represents a magnetic field and μ is the *magnetic permeability*:

$$\begin{cases} curl(H) = J \text{ in } \Omega \\ div(\mu H) = 0 \text{ in } \Omega \end{cases} \tag{1.1}$$

From mathematical analysis (see e.g. [1]), we know that the divergence-free nature of μH determines the existence of a *vector potential* $\boldsymbol{\phi}$, such that $\mu H = curl(\boldsymbol{\phi})$. Hence, problem (1.1) becomes a *vector potential problem*:

$$\begin{cases} \mathbf{curl}(\mu^{-1} curl(\boldsymbol{\phi})) = J \text{ in } \Omega \\ \boldsymbol{\phi} \cdot \mathbf{t} = 0 \text{ on } \partial\Omega \end{cases} \tag{1.2}$$

The object of our study is the eigenvalue problem associated to (1.2) in weak form. Note that, in order to uniquely specify ϕ, problem (1.2) is usually completed with the *Gauge condition*:

$$\int_\Omega \boldsymbol{\phi} \cdot \nabla q = 0 \quad \forall q \in H^1_0(\Omega)$$

This will actually be done, for this problem, in Section 1.5.

1.2 The problem

For the sake of lightening the notation, we introduce the following Hilbert spaces:

$$H(div^0, \Omega) = \{\mathbf{v} \in [L^2(\Omega)]^2 : div(\mathbf{v}) = 0\}$$
$$H_0(curl; \Omega) = \{\mathbf{v} \in [L^2(\Omega)]^2 : curl(\mathbf{v}) \in L^2(\Omega), \mathbf{v} \cdot \mathbf{t}|_{\partial\Omega} = 0\}$$

The problem at hand then reads: find $\mathbf{u} \in H_0(curl; \Omega) \setminus \{0\}$ and $\omega \in \mathbb{R}$ such that:

$$\int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v}) = \omega^2 \int_\Omega \mathbf{u} \cdot \mathbf{v} \qquad \forall \mathbf{v} \in H_0(curl; \Omega) \tag{1.3}$$

Note that this is an eigenvalue problem *in weak form*, since it has the structure:

$$< T\mathbf{u}, \mathbf{v} >= \omega^2 < \mathbf{u}, \mathbf{v} > \quad \forall \mathbf{v} \in H_0(curl; \Omega)$$

where $< \mathbf{u}, \mathbf{v} >= \int_\Omega \mathbf{u} \cdot \mathbf{v}$ and T is the operator associated to the problem, implicitly defined as:

$$< T\mathbf{u}, \mathbf{v} >= \int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v}) \quad \forall \mathbf{v} \in H_0(curl; \Omega)$$

With abuse of notations, we would write "$T\mathbf{u} = \mathbf{curl}(curl(\mathbf{u}))$".

1.3 THE EIGENVALUE $\omega^2 = 0$ AND ITS EIGENSPACE

This section contains the answer to question 1 of the assignment.
In order to prove that $\omega^2 = 0$ is an eigenvalue for (1.3), we need to show that:

$$\exists \mathbf{u} \in H_0(curl;\Omega) \setminus \{0\} : \int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v}) = 0 \quad \forall \mathbf{v} \in H_0(curl;\Omega) \tag{1.4}$$

Clearly, if $curl(\mathbf{u}) = 0$, (1.4) holds. Therefore, introducing the same notation as [3], we define:

$$H_0(curl^0;\Omega) = \{\mathbf{v} \in H_0(curl;\Omega) : curl(\mathbf{v}) = 0\}$$

Thus, if **u** is a non-zero function in $H_0(curl^0;\Omega)$[1], it verifies (1.4) and we can conclude that $\omega^2 = 0$ is an eigenvalue for (1.3).

Now we prove that the zero-curl condition is also necessary for **u** to be an eigenfunction relative to $\omega^2 = 0$, i.e that the eigenspace for 0 is in fact $H_0(curl^0;\Omega)$. Choosing $\mathbf{v} = \mathbf{u}$ (note that **v** can be chosen arbitrarily in $H_0(curl;\Omega)$) in (1.4), we obtain:

$$\int_\Omega (curl(\mathbf{u}))^2 = 0$$

Hence, if there exists a non-zero measure subset of Ω on which $curl(\mathbf{u}) \neq 0$, we get a contradiction. Therefore, we can conclude that the eigenspace relative to $\omega^2 = 0$, which will be denoted as E_0, coincides with $H_0(curl^0;\Omega)$.

In order to characterize this space, we prove the result (stated in [4]) that curl-free functions can be expressed as gradients of a potential in $H_0^1(\Omega)$:

$$H_0(curl^0;\Omega) = \nabla(H_0^1(\Omega)) \tag{1.5}$$

The proof relies on the following theorem:

Theorem 1 (Helmholtz-Weyl). *$L^2(\Omega) = G_1 \oplus G_2 \oplus G_3$, where:*

$$G_1 = \{f \in L^2(\Omega) : div(f) = 0, \gamma_\nu f = 0\}$$

$$G_2 = \{f \in L^2(\Omega) : div(f) = 0, \exists g \in H^1(\Omega) : f = \nabla g\}$$

$$G_3 = \{f \in L^2(\Omega) : \exists g \in H_0^1(\Omega) : f = \nabla g\}$$

Note that, since $G_3 = \{f \in L^2(\Omega) : \exists g \in H_0^1(\Omega) : f = \nabla g\}$, it can be denoted as $\nabla(H_0^1(\Omega))$.

Proof. (of Equation 1.5)
We will prove the inclusions:

1. $H_0(curl^0;\Omega) \subset \nabla(H_0^1(\Omega))$
2. $H_0(curl^0;\Omega) \supset \nabla(H_0^1(\Omega))$

Which, combined together, give the desired result.

[1]The fact that this space is not trivial will be proven later on in the section, with its characterization.

1. Let $\mathbf{v} \in H_0(curl^0;\Omega)$. By definition, $\mathbf{v} \in (L^2(\Omega))^2$, $curl(\mathbf{v}) = 0$ and $(\mathbf{v} \cdot \mathbf{t})|_{\partial\Omega} = 0$.
   Since $\mathbf{v} \in (L^2(\Omega))^2$, by Theorem 1 we can write $\mathbf{v}$ as a sum of a divergence-free function and a curl-free function. Highlighting the three different components of the Helmhotz decomposition of $\mathbf{v}$, we get:
$$\mathbf{v} = f_1 + f_2 + f_3 \quad f_i \in G_i$$
   Then, applying the curl, which is a linear operator, to the above equation, we obtain:
$$0 = curl(\mathbf{v}) = curl(f_1) + curl(f_2) + curl(f_3)$$
   Since f_2 and f_3 represent the curl-free part, we have that necessarily $curl(f_1) = 0$. Moreover, since $f_1 \in G_1$ we have that $div(f_1) = 0$.
   The same holds for f_2: it is both divergence and curl free. Therefore, $f_1 + f_2$ is both curl and divergence free, so that this sum belongs to $H_0(curl;\Omega) \cap H(div^0;\Omega)$, because $((f_1+f_2)\cdot\mathbf{t})|_{\partial\Omega} = 0$ as well[2].
   By applying the result in [6] for which a function in $H_0(curl;\Omega) \cap H(div^0;\Omega)$ with zero-curl is the zero function (recall that Ω is simply connected by hypothesis), we get that $f_1 + f_2 = 0$, hence $\mathbf{v} = f_3 \in G_3 = \nabla(H_0^1(\Omega))$, and this proves the desired inclusion.
2. Let $u \in H_0^1(\Omega)$, and consider ∇u. We prove that $\nabla u \in H_0(curl^0;\Omega)$:
   - $\nabla u \in (L^2(\Omega))^2$ since $u \in H_0^1(\Omega)$;
   - $curl(\nabla u) = 0$ since the curl of the gradient of a scalar is always zero;
   - $(\nabla u \cdot \mathbf{t})|_{\partial\Omega} = 0$ because from the *Gradient's Rule* (see [7]) we have that:
$$\nabla u \cdot \mathbf{t}|_{\partial\Omega} = \left.\frac{\partial u}{\partial \mathbf{t}}\right|_{\partial\Omega}$$
     Since, $u \in H_0^1$, the value on the boundary is constantly equal to zero (so there is no change along the tangential component). Therefore the tangential derivative of u on the boundary, vanishes and $\nabla u \cdot \mathbf{t}|_{\partial\Omega} = 0$.

Hence, we have managed to prove that $\nabla u \in H_0(curl^0;\Omega)$, which is the second desired inclusion.
This ends the proof. □

1.4 THE NON-ZERO EIGENVALUES AND THEIR EIGENSPACES

This section contains the answers to questions 2 and 3 of the assignment.
We now focus on the non-zero eigenvalues. The eigenvalue problem reads:
find $\mathbf{u} \in H_0(curl;\Omega) \setminus \{0\}$, $\omega \in \mathbb{R} \setminus \{0\}$ such that
$$\int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v}) = \omega^2 \int_\Omega \mathbf{u} \cdot \mathbf{v} \quad \forall \mathbf{v} \in H_0(curl;\Omega) \tag{1.6}$$

First of all, we recall a result from functional analysis: indicating with H a separable Hilbert space, and with $\mathcal{B}(H)$ the set of linear and bounded operators on H, we have that:

Theorem 2. *If $T \in \mathcal{B}(H)$ is symmetric, then the eigenvectors associated with distinct eigenvalues are orthogonal (so they're linearly independent). Therefore the eigenvalues of T are at most countable.*

[2]Note that $(\mathbf{v} \cdot \mathbf{t})|_{\partial\Omega} = 0$ (by hypothesis) and $(f_3 \cdot \mathbf{t})|_{\partial\Omega} = 0$.

Note that since $H = H_0(curl;\Omega)$ is a Hilbert subspace of $L^2(\Omega)$, which is separable, it is separable as well. Now, recall that the operator T is represented by the weak formulation of the curl of curl. It is:

- **symmetric**:

$$< T\mathbf{u}, \mathbf{v} >= \int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v}) = \int_\Omega curl(\mathbf{v}) \cdot curl(\mathbf{u}) =< \mathbf{u}, T\mathbf{v} > \quad \forall \mathbf{u}, \mathbf{v} \in H_0(curl;\Omega)$$

- **bounded**; we apply the equivalent definition of operator norm for symmetric operators:

$$||T||_{\mathcal{B}(H)} = \sup_{||\mathbf{u}||_H=1} | < T\mathbf{u}, \mathbf{u} > | \quad \forall \mathbf{u} \in H$$

  In our case, the natural norm on the space H is $||\mathbf{u}||_{H(curl)} = \sqrt{||\mathbf{u}||^2_{L^2} + ||curl(\mathbf{u})||^2_{L^2}}$. Hence, we can prove boundedness:

$$| < T\mathbf{u}, \mathbf{u} > | = \int_\Omega |curl(\mathbf{u})|^2 = ||curl(\mathbf{u})||^2_{L^2} \leq ||\mathbf{u}||_{H(curl)} \quad \forall \mathbf{u} \in H$$

  We get that $||T||_{\mathcal{B}(H)} \leq 1 < +\infty$ and the operator is bounded.

Therefore we can apply Theorem 2 and conclude that the eigenspace associated to the eigenvalue $\omega^2 \neq 0$, denoted as E_ω, is such that:

$$E_0 \perp E_\omega \quad \forall \omega \neq 0$$

Now, we will again exploit the Helmholtz decomposition: recalling that E_0 coincides with G_3, we have that:

$$\forall \omega \in \mathbb{R} \setminus \{0\} \quad \mathbf{v} \in E_\omega \Rightarrow \mathbf{v} \in (E_0)^\perp \iff \mathbf{v} \in (G_3)^\perp \iff \mathbf{v} \in G_1 \oplus G_2$$

In particular, from their definition we see that the spaces G_1 and G_2 are divergence-free: therefore, each eigenvalue associated to $\omega \neq 0$ is divergence-free.

Having proved that eigenvectors associated to non-zero eigenvalues are divergence-free, we can rewrite (1.6) as follows: find $\mathbf{u} \in X(\Omega) := H_0(curl;\Omega) \setminus \{0\} \cap H(div^0;\Omega)$ and $\omega \in \mathbb{R} \setminus \{0\}$ such that:

$$\int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v}) = \omega^2 \int_\Omega \mathbf{u} \cdot \mathbf{v} \quad \forall \mathbf{v} \in X(\Omega) \tag{1.7}$$

Now, we will prove that there exists a diverging sequence of positive eigenvalues $\{\omega_i\}_{i=1}^N$ and that each non-zero eigenvalue has a finite dimensional eigenspace.

In order to prove this fact, we will exploit the following result:

Theorem 3 (Spectral Theorem for compact, self-adjoint operators)**.** *Let H be a separable Hilbert space. Let S be a symmetric and compact operator on H. Then $\sigma(S)$ (i.e. the spectrum of S) is either finite or is a sequence $\{\lambda_n\}_{n=1}^\infty$ which converges to 0 and 0 can be either an eigenvalue or not. Moreover, the eigenvectors can always be arranged in such a way that they form an o.n.b. of H.*

First, note that our solution space $X(\Omega)$ is a separable Hilbert space. In fact, both $H_0(curl;\Omega)$ and $H(div^0;\Omega)$ are Hilbert spaces so they are closed in $(L^2(\Omega))^2$ and their intersection is closed as well. Since a closed subspace of a Hilbert space is a Hilbert space, we deduce that $X(\Omega)$ is a Hilbert space. Moreover, because $X(\Omega)$ is compactly embedded in $(L^2(\Omega))^2$, it is separable by heredity.

Note also that on $X(\Omega)$ a norm equivalent to $||\cdot||_{H(curl)}$ can be adopted: $||curl(\cdot)||_{L^2}$. It is trivial to prove that $||curl(\mathbf{u})||_{L^2} \leq ||\mathbf{u}||_{H(curl)}$ $\forall \mathbf{u} \in X(\Omega)$, and, thanks to the fact that

$$\exists C > 0 : ||\mathbf{u}||_{L^2} \leq C||curl(\mathbf{u})||_{L^2} \quad \forall \mathbf{u} \in X(\Omega)$$

we get the equivalence of the norms.

Now, we prove the desired result for Equation 1.7. The proof is inspired by what is done in [5] (Theorem 9.31) for the operator representing the laplacian, but developing all the reasoning for our operator T, namely the curl of the curl.
Note that, thanks to the following identity that holds for all vector fields $\mathbf{A}$:

$$curl(curl(\mathbf{A})) = \nabla(div\mathbf{A}) - \Delta^2\mathbf{A}$$

combined with the fact that $div(\mathbf{u}) = 0$ for the eigenvectors that we are dealing with, we can also establish an identification between the operators T and $-\Delta$.

Proof. Let us denote by H the Hilbert space $X(\Omega)$.
Let $\mathbf{f} \in (L^2(\Omega))^2$ and consider the following problem: find $\mathbf{u} \in H$ s.t.

$$\int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v} \quad \forall \mathbf{v} \in H \tag{1.8}$$

Let us denote by S the operator which goes from $(L^2(\Omega))^2 \to (L^2(\Omega))^2$ and associates to $\mathbf{f}$ the solution of (1.8): $S\mathbf{f} = \mathbf{u}$. Let us now verify the Lax-Milgram assumptions. Defining the bilinear form $a(\mathbf{u}, \mathbf{v}) := \int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v})$, we check that it is:

1. *Symmetric*: trivially,

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v}) = \int_\Omega curl(\mathbf{v}) \cdot curl(\mathbf{u}) = a(\mathbf{v}, \mathbf{u}) \quad \forall \mathbf{u}, \mathbf{v} \in X(\Omega)$$

2. *Continuous*: using Hölder's inequlity,

$$|a(\mathbf{u}, \mathbf{v})| \leq \int_\Omega |curl(\mathbf{u}) \cdot curl(\mathbf{v})| \leq ||curl(\mathbf{u})||_{L^2}||curl(\mathbf{v})||_{L^2} \quad \forall \mathbf{u}, \mathbf{v} \in X(\Omega)$$

3. *Coercive*:

$$a(\mathbf{u}, \mathbf{u}) = \int_\Omega (curl(\mathbf{u}))^2 = ||curl(\mathbf{u})||^2_{L^2} \quad \forall \mathbf{u} \in X(\Omega)$$

Now, the operator $S : (L^2(\Omega))^2 \to (L^2(\Omega))^2$ is symmetric and compact, because $H \subset (L^2(\Omega))^2$ with compact embedding. Moreover, $Ker(S) = \{0\}$ and $< S\mathbf{f}, \mathbf{f} >_{L^2} \geq 0 \quad \forall \mathbf{f} \in L^2$, because:

$$< S\mathbf{f}, \mathbf{f} >_{L^2} = \int_\Omega |curl(\mathbf{f})|^2 \geq 0 \quad \forall \mathbf{f} \in L^2$$

Hence, we can apply a corollary to Theorem 3 (Theorem 6.11 of [5]) and conclude that $\exists \{\mu_n\}_{n\in\mathbb{N}}$, a countable sequence of eigenvalues of S such that $\mu_n > 0\ \forall n$, $\mu_n \to 0$ as $n \to \infty$.
Denoting by $\mathbf{e}_n$ the eigenvector associated to μ_n, we have:

$$\int_\Omega curl(\mathbf{e}_n) \cdot curl(\mathbf{v}) = \frac{1}{\mu_n} \int_\Omega \mathbf{e}_n \cdot \mathbf{v} \quad \forall \mathbf{v} \in H \tag{1.9}$$

Therefore, the target sequence of eigenvalues of (1.7) is given by the countable set $\{\omega_n^2\}_{n\in\mathbb{N}}$, where $\omega_n^2 = \frac{1}{\mu_n}$ and clearly $\omega_n^2 > 0$, $\omega_n^2 \to \infty$ as $n \to \infty$. $\square$

We are now left to discuss the dimension of the eigenspaces associated to the non-zero eigenvalues. This can be proved by applying the following theorem to operator S of the previous proof, which shares eigenspaces with our problem:

Theorem 4. *Let H be a separable Hilbert space, and T a compact operator on H. If $\lambda \neq 0$ is an eigenvalue for T, then E_λ is finite dimensional.*

Proof. We proceed by contradiction: let $\{v_n\}_{n\in\mathbb{N}}$ an orthonormal (infinite) basis of E_λ. Then, by a corollary of Parseval inequality, $v_n \rightharpoonup 0$. Since $Tv_n = \lambda v_n$, we have $Tv_n \to 0$ (because T is compact, hence *weak-strong continuous*), but $v_n \nrightarrow 0$, which contradicts the continuity of T. □

By applying this result, we deduce that each non-zero eigenvalue has a finite dimensional eigenspace.

1.5 MIXED FORMULATION

This section contains the answer to question 4 of the assignment.
We will now analyze the relation between problem (1.3) and the following mixed formulation:

$$\begin{cases} \int_\Omega curl(\mathbf{u}) \cdot curl(\mathbf{v}) + \int_\Omega \mathbf{v} \cdot \nabla p = \omega^2 \int_\Omega \mathbf{u} \cdot \mathbf{v} \quad \forall \mathbf{v} \in H_0(curl;\Omega) \\ \int_\Omega \mathbf{u} \cdot \nabla q = 0 \quad \forall q \in H_0^1(\Omega) \end{cases} \tag{1.10}$$

Recall the *Gauge condition* previously defined in Section 1.1:

$$\int_\Omega \mathbf{u} \cdot \nabla q = 0 \quad \forall q \text{ in } H_0^1(\Omega)$$

Equation 1.10 can be interpreted as the addition of this condition to (1.3). We will now prove that (1.3) and (1.10) share the same non-zero eigenvalues. Note that, for the case of E_0, the following reasoning is not valid, since it is based on the divergence-free nature of the functions in the eigenspace, which is not true in E_0. Hence, for the development of the proof, we set ourselves in $X(\Omega)$.

Proof. On the one hand, let $(\mathbf{u}, p, \omega^2)$ be a solution of the coupled problem. If $\mathbf{v} \in X(\Omega)$, then by integration by parts:

$$\int_\Omega \mathbf{v} \cdot \nabla p = -\int_\Omega div(\mathbf{v}) p = 0$$

Therefore $\mathbf{u}$ satisfisfies exactly (1.3); hence, all the non-zero eigenvalues of the coupled problem are also eigenvalues of the original problem.

On the other hand, let us start from (1.3) and consider the eigenvectors related to the non-zero eigenvalues. From section 1.4 we know that they are divergence-free functions, hence, by integration by parts and using that $q \in H_0^1(\Omega)$:

$$\int_\Omega \mathbf{u} \cdot \nabla q = -\int_\Omega div(\mathbf{u}) q = 0 \quad \forall q \in H_0^1(\Omega)$$

Therefore, the *Gauge condition* is verified and the second equation of the coupled problem (1.10) is obtained. Now, we need to find a p such that $\int_\Omega \nabla p \cdot \mathbf{v} = 0 \quad \forall \mathbf{v} \in H_0(curl;\Omega)$ to recover (1.10). The only suitable p that verifies the above condition is $p = 0$. In fact, as seen in the lectures, it can be verified by choosing $\mathbf{v} = \nabla p$ [3] in the first equation of (1.10).

□

[3]Note that this is a legitimate choice, because we know that the gradient of a H_0^1 function is in $H_0(curl)$.

CHAPTER 2: NUMERICAL SOLUTION

We now proceed to the numerical discretization of (1.3). Namely, we first introduce a discrete partition $\mathcal{T}_h$ on $\Omega = [0, \pi]^2$, then we set ourselves in the finite-dimensional space $X_h \subset X(\Omega)$, and look for $u_h \in X_h$, $\omega_h \in \mathbb{R}$ such that:

$$\int_\Omega curl(\mathbf{u}_h) \cdot curl(\mathbf{v}_h) = \omega_h^2 \int_\Omega \mathbf{u}_h \cdot \mathbf{v}_h \quad \forall \mathbf{v}_h \in X_h \tag{2.1}$$

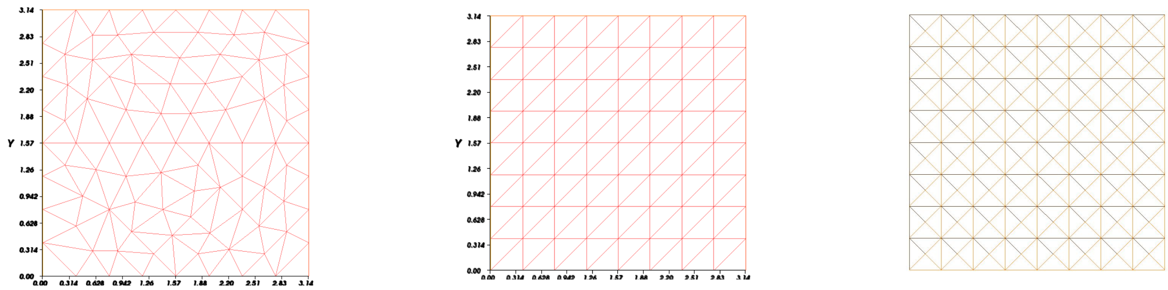
We are now left to discuss the choice of the partition and of the finite dimensional space.

2.1 BUILDING THE MESH $\mathcal{T}_h$

We approached the numeric problem with three different meshes.

- An **unstructured mesh**; note that, as the domain is symmetric and does not present critical points (such as holes), we used a mesh with comparable element size in each part of Ω (see Figure 2.1a).
- A **triangular mesh**, the one generated by the standard `square` command in `FreeFem++` (see Figure 2.1b).
- A **criss-cross mesh**: to generate this mesh (in Figure 2.1c), we cannot rely on `FreeFem++` built-in commands. Hence, we used `FeNiCs`, where a criss-cross mesh can be generated with:

```
mesh = RectangleMesh(Point(0, 0), Point(pi, pi), N, N, "crossed")
```



(A) Unstructured mesh (B) Triangular mesh (C) Criss-cross mesh

FIGURE 2.1
The three types of mesh employed, with 3 refinements

Starting from a mesh where each edge is split in 4, we performed some refinements in order to find a suitable mesh resolution, as will be discussed in chapter 3.

2.2 CHOICE OF X_h

We analyzed two different choices of X_h:

- $X_h = (\mathbb{P}_1^C)^2$, which corresponds to the standard discretization with piecewise continuous linear polynomials:
$$X_h = \{\mathbf{v} \in X(\Omega) \cap \mathcal{C}(\Omega) : \mathbf{v}|_K \in (\mathbb{P}_1)^2 \quad \forall K \in \mathcal{T}_h\}$$

- **Nédélec elements** ($\mathbb{N}_k$), also called *edge elements*, which basically consist of turned Raviart-Thomas elements ($\mathbb{RT}_k$):
$$\mathbb{N}_k = \{\boldsymbol{\sigma} \times \boldsymbol{n}_z, \text{ where } \boldsymbol{\sigma} \in \mathbb{RT}_k\}$$
We recall that $\mathbb{RT}_k(K) = ((\mathbb{P}_k(K))^2 \oplus \mathbf{x}\widetilde{\mathbb{P}}_k(K))$, where $\widetilde{\mathbb{P}}_k(K)$ denotes the *scalar homogeneous* polynomials of degree k on K, hence this space contains *some* polynomials of degree $k+1$, but does not coincide with the whole $(\mathbb{P}_{k+1}(K))^2$.
Introducing the notation $\mathbf{x}^{\perp} := [-y, x]^T$ to indicate the rotation of $\mathbf{x}$, we get:
$$\mathbb{N}_k(K) = ((\mathbb{P}_k(K))^2 \oplus \mathbf{x}^{\perp}\widetilde{\mathbb{P}}_k(K))$$
Indeed, when performing the rotation of $\mathbb{RT}$ elements, the only vector that is rotated is $\mathbf{x}$, as $(\mathbb{P}_k(K))^2$ and $\widetilde{\mathbb{P}}_k(K)$ already represent the whole polynomial spaces.
Let us denote with τ the rotated vector $\boldsymbol{\sigma} \times \mathbf{n}_z$; for the polynomial degree k, the degrees of freedom of the $\mathbb{N}_k$ elements can be determined simply by rotating the ones of the $\mathbb{RT}_k$ elements:
  1. $\int_e (\boldsymbol{\sigma} \cdot \mathbf{n}_e) p_e \to \int_e (\boldsymbol{\tau} \cdot \mathbf{t}_e) p_e \quad \forall p_e \in \mathbb{P}_k(e)$ where e indicates an internal[1] edge of the mesh
  2. $\int_e \boldsymbol{\sigma} \to \int_e \boldsymbol{\tau}$

  We know that this choice of degrees of freedom is suitable because it guarantees *inter-element continuity* of the tangential component and they are *unisolvable*. This means that putting all of our degrees of freedom equal to 0, we obtain the zero element.

2.3 BOUNDARY CONDITIONS

For the assignment of the boundary conditions, the implementation of Nédélec Elements in `FreeFem++` forces us to impose a condition on each component of the solution on each boundary. Hence, we could not apply just the condition of the null tangential component on the boundary. We proceeded in two ways:

- *Imposition of the exact solution* on $\partial\Omega$: this strategy relies upon the knowledge of the exact solution, that is, in our simple $2D$ case, $\mathbf{u} = (sin(mx), sin(ny))^T$ (see [2]).
We have performed two `for` loops and computed one eigenvalue per iteration, imposing the exact solution on $\partial\Omega$.

- *Homogeneous Dirichlet boundary conditions*: we actually noted that the solution for the eigenvalues was independent on the boundary conditions for $\mathbf{u}$. Hence, we have imposed a zero solution (both components) on the whole boundary and computed all the desired spectrum in one iteration.

[1]In our case, the tangential component of the function is zero on the mesh boundary.

2.4 Numerical Method

In order to solve the eigenvalue problem in `FreeFem++`, we used the `Eigenvalue` command with the *shift-invert method*. The original system $Au = \lambda Bu$ is transformed into a new one of the form $(A - \sigma B)w = Bu$, where σ is called the *shift*. This value indicates that we are looking for the N ($N = 48$ in our case) eigenvalues that are the closest to σ.
The values of σ were chosen in order to obtain the eigenvalues we were interested in, namely the non-zero eigenvalues starting from $\omega^2 = 1$. Knowing the exact eigenvalues, we could compute the mean of the first 48 ones and get an initial guess. Then, it was fine-tuned by hand. Choosing a value too small resulted in some zero eigenvalues (that we do not want to detect in this study[2]), while choosing a bigger value resulted in the exclusion of the first non-zero eigenvalues. Different values were found for the different meshes and element types, as described in the next section.

Chapter 3: Results

This chapter contains the answer to question 5 of the assignment.
We now report the results obtained for the eigenvalues ω^2 and compare them with the exact solution known from the theory, $\omega^2 = n^2 + m^2$ (where $n, m = 0, 1, \dots$) for each choice of mesh and finite dimensional space.
We studied meshes with different resolutions, with number of refinements ranging from 3 to 8; we first report the error trend as a function of the refinement for the case of Nédélec Elements on Unstructured Mesh (Figure 3.1). The error was computed using the $||\cdot||_\infty$ norm for the eigenvalues. For the case of 7 refinements we obtain an error below our tolerance, which has been fixed to $2 \cdot 10^{-2}$; this will be the resolution adopted to obtain the following results.

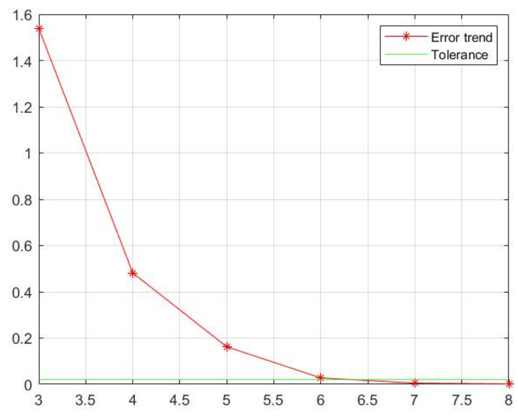


Figure 3.1
Trend of the error in $||\cdot||_\infty$ for different mesh refinements.

[2]Indeed, we know from the theoretical study of the problem that the eigenspace E_0 is infinite-dimensional, hence we expect to find that 0 is an eigenvalue with infinite multiplicity.

3.1 $\mathbb{P}_1^C$ ELEMENTS

Mesh	$\omega^2 = 1$	$\omega^2 = 2$	$\omega^2 = 4$	$\omega^2 = 5$	$\omega^2 = 8$	$\omega^2 = 9$	**Error in** $\|\|\cdot\|\|_\infty$
Unstructured	23.439	23.4976	23.5214	23.5587	23.6124	23.6192	22.483
Triangular	26.0894	26.1711	26.182	26.1859	26.2099	26.2584	25.137
Criss-cross	1.000017	2.000067	4.000268	5.000418	5.999699	8.001071	7.9886

TABLE 3.1
Results for the first (different) 6 non-zero eigenvalues, with $\mathbb{P}_1^C$ Elements and grid with 7 refinements

For the unstructured and the triangular case, we note that all the values computed (and not only the 6 reported in Table 3.1) remain stuck around the values 23 and 26, which are the two values of the *shift* σ used for these cases. Focusing on the criss-cross mesh results, note that we obtain a *spurious* mode $\omega^2 = 6$, which is not present in the exact spectrum, and if we compute more eigenvalues, we get other spurious eigenvalues (see Figure 3.2). Hence, the $\mathbb{P}_1^C$ elements are not suitable to discretize the Maxwell eigenvalue problem.

3.2 NÉDÉLEC ELEMENTS

Mesh	$\omega^2 = 1$	$\omega^2 = 1$	$\omega^2 = 2$	$\omega^2 = 4$	$\omega^2 = 4$	$\omega^2 = 5$	**Error in** $\|\|\cdot\|\|_\infty$
Unstructured	1.00000	1.00000	2.00000	3.99999	3.99999	4.99997	0.0024
Triangular	0.99997	0.999997	2.00003	3.99973	3.99973	4.99973	0.0241
Criss-cross	1.00000	1.00000	1.99997	4.00007	4.00007	4.99990	0.0130

TABLE 3.2
Results for the first 6 non-zero eigenvalues, with Nédélec Elements and grid with 7 refinements

In Table 3.2 we have reported the results for the Nédélec elements, on a grid with 7 refinements (each edge is partitioned into $2^7 = 128$ subedges). On this high-resolution grid, the results turn out to be extremely accurate for each of the proposed meshes, with an absolute error of the order 10^{-2} even for the worse performing mesh (in this case, the triangular one). The errors in the discrete ∞ norm[1] state that the best results are provided by the unstructured mesh. These are plotted in Figure 3.3.

3.3 CONCLUSION

In this project, we studied the eigenvalues of the Maxwell cavity. Starting from the variational form, we charaterized the eigenspace corresponding to the zero eigenvalue and proved that it corresponds to the space $\nabla(H_0^1(\Omega))$. Then, we studied the eigenvectors corresponding to the non-zero ones, showing that they are divergence-free and their eigenspace is finite-dimensional. Finally, we described an equivalent mixed formulation introducing the *Gauge condition*. With these results in mind, we moved on to the numerical approximation of the problem, using different types of meshes (Unstructured, Triangular and Criss-cross) and elements ($\mathbb{P}_1^C$ and Nédélec). After convergence tests to find a suitable refinement, the first 48 non-zero eigenvalues were computed for each method. Comparison studies revealed that the Nédélec elements with Unstructured mesh was the most accurate method, while others presented bigger errors, and sometimes spurious modes.

[1]Given a vector $\mathbf{v} \in \mathbb{R}^n$, $||\mathbf{v}||_\infty := max_{i=1,\ldots,n}|v_i|$

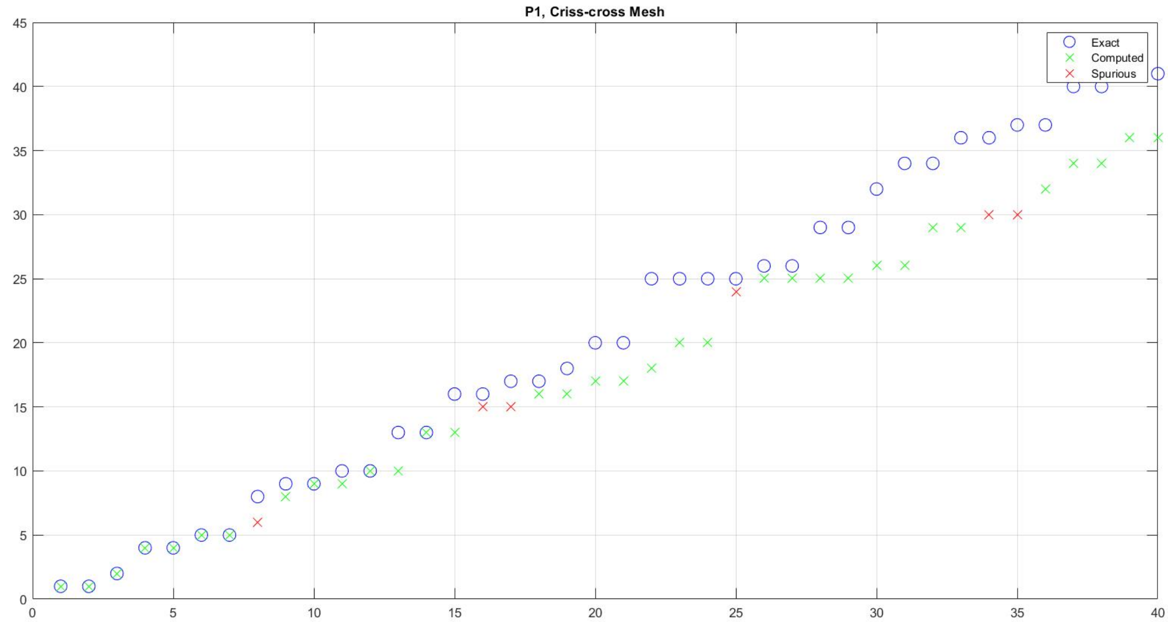


FIGURE 3.2
Criss-cross mesh (7 refinements), $\mathbb{P}_1^C$ elements

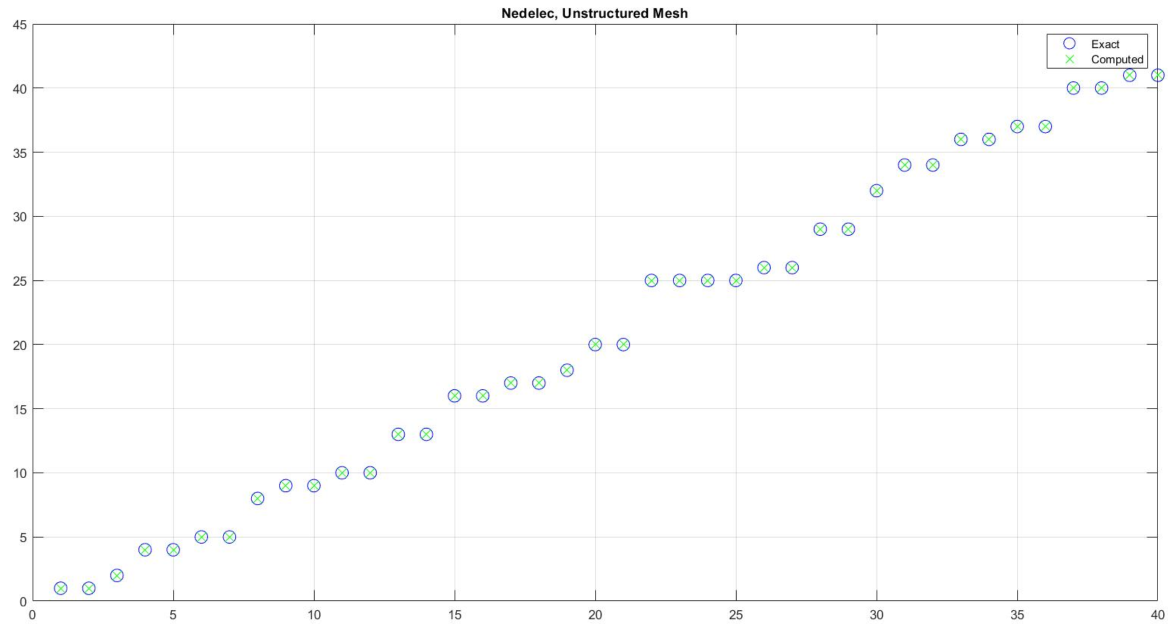


FIGURE 3.3
Unstructured mesh (7 refinements), Nédélec elements

Bibliography

[1] C. Amrouche, C. Bernardi, M. Dauge, and V. Girault. Vector potentials in three-dimensional non-smooth domains. *Mathematical Methods in the Applied Sciences*, 21(9):823–864, 1998.

[2] DN. Arnold. Differential complexes and numerical stability. *Proceedings of the ICM*, 1:137–157, 2002.

[3] D. Boffi, P. Fernandes, L. Gastaldi, and Perugia I. Computational models of electromagnetic resonators: analysis of edge element approximation. *SIAM Journal on Numerical Analysis*, 36(4):1264–1290, 1999.

[4] D. Boffi, L. Gastaldi, R. Rodriguez, and I. Sebestova. A posteriori error estimates for Maxwell's eigenvalue problem. *Journal of Scientific Computing*, 78(2):1250–1271, 2019.

[5] H. Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer, 2010.

[6] V. Girault and P. Raviart. *Finite Element Methods for Navier Stokes Equations, Theory and Algorithms*. Springer-Verlag, 1986.

[7] CD. Pagani and S. Salsa. *Analisi Matematica II*. Zanichelli, 2016.

Appendix A: FreeFem++ Code

We report the core part of the code to run the convergence test; note that in the comments we have reported the commands to run it with different mesh and element type.

```
//Number of refinements
int I = 7;
for(int i = 3; i <= I; i++)
{
    //Build mesh
    border l1(t=a,b){x=t;y=0;label=1;}
    border l2(t=a,b){x=b;y=t;label=2;}
    border l3(t=a,b){x=b-t;y=b;label=3;}
    border l4(t=a,b){x=a;y=b-t;label=4;}
    mesh Th=buildmesh(l1(2^i)+l2(2^i)+l3(2^i)+l4(2^i));
    //mesh Th = square(2^i, 2^i, [x0+(x1-x0)*x, y0+(y1-y0)*y]); Triangular mesh
    //FEspace
    fespace Vh(Th, RT0Ortho);
    Vh [u1, u2], [v1, v2];
    //fespace Vh(Th, [P1,P1]); P1 elements
    //Vh [u1, u2], [v1, v2];
    //Solve the problem for each valid couple (n,m)
    for(int n = 0; n < N; n++)
    {
        for(int m = 0; m < N; m++)
        {
            omega2 = n*n + m*m;
            if(omega2 != 0) //Don't consider the zero eigenvalue
            {
                //Compute the new shift
                sigma = omega2 + tolerance;
                //Setup problem in variational form using the Shift-invert method
                varf op([u1, u2], [v1, v2]) = int2d(Th)(dx(u2)*dx(v2)
                                            - dy(u1)*dx(v2) - dx(u2)*dy(v1)
                                            + dy(u1)*dy(v1))
                                            - int2d(Th)(sigma * (u1*v1 + u2*v2))
                                            + on(1, 3, u1=sin(m*x), u2=sin(n*y))
                                            + on(2, 4, u1=sin(m*x), u2=sin(n*y));
                varf bb([u1, u2], [v1, v2])  = int2d(Th)(u1*v1 + u2*v2);
                //Translate into matrix problem and specify solvers
                //OP = A - sigma B and we want to solve OPw = Bu
                matrix OP = op(Vh, Vh, solver=Crout, factorize=1);
                matrix B  = bb(Vh, Vh, solver=CG, eps=1e-20);
                //Solve the eigenvalue problem, store the number of eigenvalues in k
                int k = EigenValue(OP, B, sym=true, sigma=sigma, value=ev,
                tol=1e-10, maxit=0, ncv=0, which="LA");
            }
        }
    }
}
```

Appendix B: FeNiCs Code

Although there is a possibility to import the mesh from `FeNiCS` to `FreeFem++` as `.msh` file (included in the script reported below), we decided to work with `FeNiCS`, as the handling of the file resulted very unpractical. Indeed, we would need to export the `.msh` file to a `.mesh` using `GMSH`, and then remove the $z-$coordinates (which are clearly all 0) with a text editor, else `FreeFem++` would not read the file correctly.
This preprocessing resulted unfeasible for the case of high-resolution mesh.

```
def eigenvalues(V, bcs):
    # Define the bilinear forms on the right- and left-hand sides
    u = TrialFunction(V)
    v = TestFunction(V)
    a = inner(curl(u), curl(v))*dx
    b = inner(u, v)*dx

    # Assemble into PETSc matrices
    dummy = v[0]*dx
    A = PETScMatrix()
    assemble_system(a, dummy, bcs, A_tensor=A)
    B = PETScMatrix()
    assemble_system(b, dummy, bcs, A_tensor=B)

    [bc.zero(B) for bc in bcs]
    solver = SLEPcEigenSolver(A, B)
    solver.parameters["solver"] = "krylov-schur"
    solver.parameters["spectral_transform"] = "shift-and-invert"
    solver.parameters["spectral_shift"] = 22 # 18.5 for P1
    neigs = 40
    solver.solve(neigs)

     # Return the computed eigenvalues in a sorted array
    computed_eigenvalues = []
    for i in range(min(neigs, solver.get_number_converged())):
        r, _ = solver.get_eigenvalue(i) # ignore the imaginary part
        computed_eigenvalues.append(r)
    return np.sort(np.array(computed_eigenvalues))

i = 7 # set refinements
N = 2**i
mesh = RectangleMesh(Point(0, 0), Point(pi, pi), N, N, "crossed")
# Mesh storage and export to .msh format
xdmf = XDMFFile("mesh.xdmf")
xdmf.write(mesh)
xdmf.close()
import meshio
meshio_mesh = meshio.read("mesh.xdmf")
meshio.write("meshio_mesh.msh", meshio_mesh)
```