

EPFL

SEMESTER PROJECT

MASTER OF SCIENCE MSc IN  
COMPUTATIONAL SCIENCE AND ENGINEERING

**Minimal Rational Interpolation for  
Time-Harmonic Maxwell's Equations**

*Fabio Matti*

supervised by  
Prof. Fabio Nobile  
Dr. Davide Pradovera

July 8, 2022

## A B S T R A C T

Minimal Rational Interpolation (MRI) provides an efficient and reliable way to approximate a meromorphic dependence of a characteristic output quantity of a model on one of its parameters. The focus of this report is on the Greedy Minimal Rational Interpolation (gMRI) algorithm and particularly on ways to enhance its performance. This algorithm is then applied to three example problems concerning the time-harmonic Maxwell's equations in the frequency domain. The advantages and disadvantages of gMRI as compared to conventional approaches for finding quantities such as resonant modes that are of interest for problems of this type are discussed. Furthermore, the application of the gMRI algorithm for finding the scattering coefficients of a Dual-Mode Circular Waveguide Filter (DMCWF) is studied.

# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Finite element discretization of the time-harmonic Maxwell's equations</b>	<b>4</b>
2.1	Vector potential formulation of the time-harmonic Maxwell's equations	4
2.2	Weak formulation for the time-harmonic potential equation . . . . .	5
2.3	Examples . . . . .	5
2.3.1	Two-dimensional resonant cavity . . . . .	6
2.3.2	Imperfect conductor . . . . .	7
2.3.3	Waveguide . . . . .	7
<b>3</b>	<b>Finite element approximation with FEniCS</b>	<b>8</b>
3.1	The Galerkin method . . . . .	8
3.2	Numerical approximation of PDEs using FEniCS . . . . .	8
<b>4</b>	<b>Minimal rational interpolation for the time-harmonic Maxwell's equations</b>	<b>11</b>
4.1	Motivation . . . . .	11
4.2	Minimal rational interpolation . . . . .	12
4.3	Greedy minimal rational interpolation . . . . .	12
4.4	Properties of the rational surrogate . . . . .	13
4.5	Finding roots of rational functions . . . . .	13
4.6	Optimization tricks for greedy minimal rational interpolation . . . . .	14
4.6.1	Additive Householder decomposition . . . . .	14
4.6.2	Stability of singular value decomposition . . . . .	15
4.6.3	Alternative representations of the surrogate . . . . .	17
<b>5</b>	<b>Examples</b>	<b>19</b>
5.1	Two-dimensional rectangular resonant cavity . . . . .	19
5.1.1	Exploration of the problem . . . . .	19
5.1.2	Numerical approximation of resonant frequencies . . . . .	20
5.1.3	Approximating solutions along a trace . . . . .	25
5.2	Imperfectly conducting boundaries . . . . .	28
5.3	Dual mode circular waveguide filter . . . . .	31
<b>6</b>	<b>Conclusion and outlook</b>	<b>35</b>

## 1 INTRODUCTION

A wide class of problems in physics and engineering concern studying the dependence of a model on one (or more) of its parameters  $\omega$ . Of interest is usually a characteristic quantity  $\mathbf{u}(\omega)$  depending on this parameter. Unless the system allows for an analytical solution, one may usually only find numerical solutions to the system for discrete values of the parameter. Minimal Rational Interpolation (MRI) offers a way to locally approximate vector fields  $\mathbf{u} : \mathbb{C} \ni \omega \mapsto \mathbf{u}(\omega)$  which exhibit a meromorphic dependence on one of their parameters. The approach has proven effective and efficient (both memory- and time-wise) in applications on Helmholtz-type problems [2, 13].

Central to this report are time-harmonic electromagnetic problems, whose parameter is the (angular) frequency. These problems are governed by the time-harmonic Maxwell's equations. Choosing the quantity of interest to be a vector potential, these equations reduce to the curl-curl equation

$$\nabla \times (\mu^{-1} \nabla \times \mathbf{u}) - \epsilon \omega^2 \mathbf{u} = \mathbf{j} \quad (1.1)$$

A justification for why a rational interpolation approach is appropriate for this class of problems will be presented in Section 4.1.

The first few pages in this report are a short guide for finding numerical solutions to the time-harmonic Maxwell's equations using the Finite Element Method (FEM). These solutions are then used in the core of this report, which gives a description of the Greedy Minimal Rational Interpolation (gMRI) algorithm. Properties of and optimization tricks for the gMRI are shown. In the end, three applications of the method are studied and discussed: the resonant modes of a two-dimensional resonant cavity, the two-dimensional cavity with an imperfectly conducting boundary, and lastly the scattering coefficients of a Dual-Mode Circular Waveguide Filter (DMCWF).

My implementations can be found on [7]. Vectors are printed in **bold** while matrices additionally have an underline. Euclidean scalar products are denoted with  $\langle \cdot, \cdot \rangle$  and norms with  $\|\cdot\|$ . The Hermitian transpose is written as  ${}^\text{H}$ . For simplicity, results are delivered without units and the angular frequency is referred to simply by frequency.

## 2 FINITE ELEMENT DISCRETIZATION OF THE TIME-HARMONIC MAXWELL'S EQUATIONS

Minimal Rational Interpolation (MRI) requires the knowledge of the solution of the problem for discrete values of the model parameter that is of interest. A way of obtaining these solutions is the Finite Element Method (FEM). For that purpose, I now derive a strong formulation for the time-harmonic Maxwell problem and subsequently convert it to its corresponding weak formulation.

### 2.1 VECTOR POTENTIAL FORMULATION OF THE TIME-HARMONIC MAXWELL'S EQUATIONS

I assume that all quantities in this section are smooth enough to perform the necessary vector calculus manipulations.

Let  $\mathbf{E}$  denote an electric field,  $\mathbf{B}$  a magnetic field strength,  $\rho$  an electric charge density, and  $\mathbf{j}$  an electric current density. Maxwell's equations are stated in [8] as

$$\langle \nabla, \epsilon \mathbf{E} \rangle = \rho \quad (2.1)$$

$$\langle \nabla, \mathbf{B} \rangle = 0 \quad (2.2)$$

$$\nabla \times \mathbf{E} = -\partial_t \mathbf{B} \quad (2.3)$$

$$\nabla \times (\mu^{-1} \mathbf{B}) = \partial_t(\epsilon \mathbf{E}) + \mathbf{j} \quad (2.4)$$

with  $\epsilon$  being the permittivity and  $\mu$  the permeability.

Equation (2.2) motivates the expression of the magnetic field  $\mathbf{B} = \nabla \times \mathbf{u}$  in terms of a vector-valued function  $\mathbf{u}$ , the vector potential (in literature commonly denoted with  $\mathbf{A}$ ). Similarly, (2.3) suggests rewriting the electric field  $\mathbf{E} = -\nabla \phi - \partial_t \mathbf{u}$  using a scalar function  $\phi$ , referred to as the scalar potential.

The physical quantities  $\mathbf{E}$  and  $\mathbf{B}$  remain unchanged if we transform  $\mathbf{u} \rightarrow \mathbf{u}' = \mathbf{u} + \nabla \psi$  or  $\phi \rightarrow \phi' = \phi - \partial_t \psi$  for an arbitrary function  $\psi$ . A convenient choice of  $\psi$  is suggested in [4] to be

$$\psi = \int_0^t \phi dt' \quad (2.5)$$

which transforms  $\phi \rightarrow \phi' = 0$  and  $\mathbf{u} \rightarrow \mathbf{u}' = \mathbf{u} + \nabla \int_0^t \phi dt'$ . Thus, the expressions for the electrical and magnetic field become

$$\mathbf{E} = -\partial_t \mathbf{u} \quad (2.6)$$

$$\mathbf{B} = \nabla \times \mathbf{u} \quad (2.7)$$

where I have subtly renamed the variable  $\mathbf{u}'$  to  $\mathbf{u}$  for simplicity.

Plugging the identities (2.6) and (2.7) into (2.4) yields (assuming a static permittivity  $\epsilon$ )

$$\nabla \times (\mu^{-1} \nabla \times \mathbf{u}) = -\epsilon \partial_t^2 \mathbf{u} + \mathbf{j} \quad (2.8)$$

For the rest of this report, I restrict myself to vector potentials  $\mathbf{u}$  that exhibit a harmonic dependence on time  $t$ , i.e. may be factorized into a term solely depending on the position  $\mathbf{x}$  and a complex exponential depending on time

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x}) \exp(i\omega t) \quad (2.9)$$

Substituting this expression into (2.8) and rearranging a little results in the

Time-harmonic potential equation

$$\nabla \times (\mu^{-1} \nabla \times \mathbf{u}) - \epsilon \omega^2 \mathbf{u} = \mathbf{j} \quad (2.10)$$

## 2.2 WEAK FORMULATION FOR THE TIME-HARMONIC POTENTIAL EQUATION

Take the inner product of equation (2.10) with an arbitrary vector-valued test function  $\mathbf{v} \in H_{\text{curl}}(\Omega)$ , where

$$H_{\text{curl}}(\Omega) = \{\mathbf{v} : \Omega \rightarrow \mathbb{C}^3, \text{ such that } \mathbf{v} \in L_2(\mathbb{C})^3, \nabla \times \mathbf{v} \in L_2(\mathbb{C})^3\} \quad (2.11)$$

and then integrate over the whole computational domain  $\Omega$  to obtain

$$\int_{\Omega} \langle \nabla \times (\mu^{-1} \nabla \times \mathbf{u}), \mathbf{v} \rangle - \omega^2 \int_{\Omega} \epsilon \langle \mathbf{u}, \mathbf{v} \rangle = \int_{\Omega} \langle \mathbf{j}, \mathbf{v} \rangle \quad (2.12)$$

Equation (2.12) may be further simplified to:

Weak formulation of the time-harmonic potential equation

$$\int_{\Omega} \langle \mu^{-1} \nabla \times \mathbf{u}, \nabla \times \mathbf{v} \rangle - \omega^2 \int_{\Omega} \epsilon \langle \mathbf{u}, \mathbf{v} \rangle = \int_{\Omega} \langle \mathbf{j}, \mathbf{v} \rangle + \int_{\partial\Omega} \underbrace{\langle (\mu^{-1} \nabla \times \mathbf{u}) \times \mathbf{n}, \mathbf{v} \rangle}_{=\mathbf{g}} \quad (2.13)$$

where  $\mathbf{n}$  denotes the outward surface normal to the boundary  $\partial\Omega$  of the computational domain  $\Omega$  and  $\mathbf{v}$  is an arbitrary element of  $H_{\text{curl}}(\Omega)$ . I allow myself to spare you the details of this computation, but nevertheless put a proper derivation in the appendix at the end of this report.

Supposing that we know the electric field  $\mathbf{E}$  along a part of the boundary  $\Gamma \subset \partial\Omega$ , we may most easily enforce it in a Dirichlet-type fashion through the relation (2.6) and the assumption (2.9)

$$\mathbf{u}|_{\Gamma} = -\frac{1}{i\omega} \mathbf{E}|_{\Gamma} \quad (2.14)$$

A known magnetic field  $\mathbf{B}$  may similarly be enforced with a Neumann-type condition following from (2.7) and (2.9)

$$\mathbf{g}|_{\Gamma} = (\mu^{-1} \mathbf{B}|_{\Gamma}) \times \mathbf{n} \quad (2.15)$$

## 2.3 EXAMPLES

I will now specialize and simplify this weak formulation for three different applications which will be studied in Section 5. To see that these problems are intimately related, I refer you to Figure 2.1.

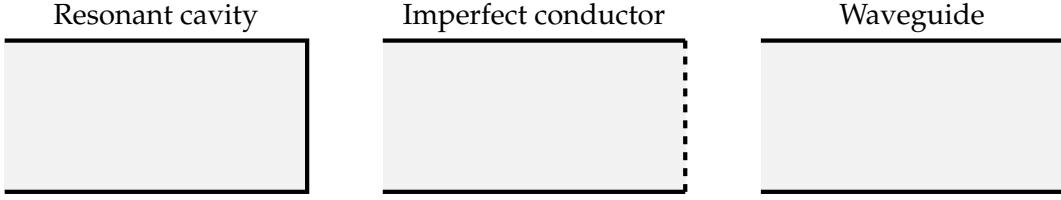


FIGURE 2.1 – Schematic visualization of the most trivial case for each of the boundary configurations that will be analyzed in Section 5. The perfectly conducting boundaries are drawn in black, while the imperfectly conducting boundary appears dashed. Inlets and outlets are left unmarked.

### 2.3.1 TWO-DIMENSIONAL RESONANT CAVITY

I refer to a resonant cavity as a region  $\Omega$  enclosed by a boundary  $\partial\Omega$ . The boundary can be subdivided into one (or more) inlets  $\Gamma_N$  and a perfectly conducting wall  $\Gamma_D = \partial\Omega \setminus \Gamma_N$  (see Figure 2.2 for an abstract visualization of such a cavity).

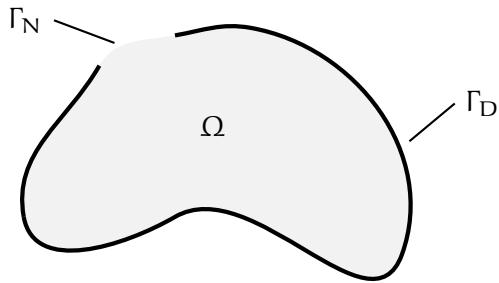


FIGURE 2.2 – An abstract example of a two-dimensional resonant cavity enclosing a domain  $\Omega$  with a perfectly conducting boundary  $\Gamma_D$  and featuring a single inlet  $\Gamma_N$ .

Now, we move down to two dimensions, i.e. assume the resonant cavity lies in a plane. Suppose the current density  $\mathbf{j} \equiv 0$  and orient the coordinate system in such a way that  $\mathbf{u} = u_z \mathbf{e}_z$  and  $\mathbf{v} = v_z \mathbf{e}_z$ . Consequently, the scalar product of the two curls in Equation (2.13) simplifies to the scalar product of two gradients:

$$\langle \mu^{-1} \nabla \times \mathbf{u}, \nabla \times \mathbf{v} \rangle = \langle \mu^{-1} \nabla u_z, \nabla v_z \rangle \quad (2.16)$$

Denote by  $g_z$  the component of  $\mathbf{g}$  in the  $z$ -direction along the inlet  $\Gamma_N$ . These simplifications allow the conversion of (2.13) into the weak formulation for a two-dimensional resonant cavity

$$\int_{\Omega} \langle \mu^{-1} \nabla u_z, \nabla v_z \rangle - \omega^2 \int_{\Omega} \epsilon u_z v_z^* = \int_{\partial\Omega} g_z v_z^* \quad (2.17)$$

for an arbitrary  $v_z \in H_1(\Omega) = \{v : \Omega \rightarrow \mathbb{C}, \text{ such that } v \in L_2(\mathbb{C}), \nabla v \in L_2(\mathbb{C})^2\}$ . Now, let  $\mathbf{E}$  and  $\mathbf{B}$  refer to the electric and magnetic fields inside the cavity. For now, I distinguish between two types of boundaries:

For the perfectly conducting boundary  $\Gamma_D$ , treated in [8], it holds that

$$\mathbf{n} \times \mathbf{E} = \mathbf{0}, \text{ on } \Gamma_D \quad (2.18)$$

For the boundaries in a two-dimensional resonant cavity (see Figure 2.2), this only holds true if  $E_z = 0$ , which translates to the Dirichlet boundary condition  $u_z|_{\Gamma_D} = 0$  in light of (2.14).

For the inlet, it is easiest to enforce the boundary condition through the magnetic field  $\mathbf{B}$  in exactly the way proposed in (2.15) (assuming  $\mathbf{n} = -\mathbf{e}_x$  as will always be the case in Section 5, cf. Figure 5.1):

$$g_z = ((\mu^{-1}\mathbf{B}) \times (-\mathbf{e}_x))_z = \mu^{-1}B_x, \text{ on } \Gamma_N \quad (2.19)$$

### 2.3.2 IMPERFECT CONDUCTOR

To simulate an imperfect boundary  $\Gamma_I$ , also called impedance boundary in literature, [8] suggests replacing the integrand  $\mathbf{g}$  that appeared in (2.13) with

$$\mathbf{g} = (\mu^{-1}\nabla \times \mathbf{u}) \times \mathbf{n} = i\omega\lambda(\mathbf{n} \times \mathbf{u}) \times \mathbf{n} \text{ on } \Gamma_I \quad (2.20)$$

with a parameter  $\lambda > 0$  which I will henceforth refer to as the impedance. Supposing that  $\mathbf{u} = u_z\mathbf{e}_z$  and only treating a two-dimensional domain, this condition simplifies to

$$g_z = i\omega\lambda u_z \text{ on } \Gamma_I \quad (2.21)$$

by using the fact that  $\mathbf{n} \perp \mathbf{u}$  and  $\|\mathbf{n}\| = 1$ , so  $(\mathbf{n} \times \mathbf{u}) \times \mathbf{n} = \mathbf{u}$ , as is demonstrated in the appendix at the end of this report.

### 2.3.3 WAVEGUIDE

Going back to (2.13) and this time staying in three dimensions, we again assume no electric current density  $\mathbf{j} \equiv 0$  is present. I suppose that the inlet is located at a constant  $x$ -value, such that the surface normal to this inlet is  $-\mathbf{e}_x$ . Conveniently, the example in Section 5.3 happens to be set up in just this way. For an incoming magnetic field at the inlet  $\Gamma_i$  with  $\mathbf{B}|_{\Gamma_i} = B_0\mathbf{e}_y$ , we see from (2.15) that this may be modelled by setting  $\mathbf{g}|_{\Gamma_i} = -\mu^{-1}B_0\mathbf{e}_z$ . At the “outlet”  $\Gamma_o$ , we set  $\mathbf{g}|_{\Gamma_o} = \mathbf{0}$ .

### 3 FINITE ELEMENT APPROXIMATION WITH FENICS

Based on the weak formulation corresponding to the time-harmonic potential equation (2.13), the Finite Element Method (FEM) can be used to approximate solutions to Equation (2.10).

#### 3.1 THE GALERKIN METHOD

It is readily seen that the weak formulation (2.13) assumes the form

$$\text{Find } \mathbf{u} \in \mathcal{H}_{\text{curl}}(\Omega), \text{ such that } \mathbf{a}_\omega(\mathbf{u}, \mathbf{v}) = \mathbf{L}(\mathbf{v}), \forall \mathbf{v} \in \mathcal{H}_{\text{curl}}(\Omega) \quad (3.1)$$

with the bilinear form

$$\mathbf{a}_\omega(\mathbf{u}, \mathbf{v}) = \underbrace{\int_{\Omega} \langle \mu^{-1} \nabla \times \mathbf{u}, \nabla \times \mathbf{v} \rangle}_{\mathbf{K}(\mathbf{u}, \mathbf{v})} - \omega^2 \underbrace{\int_{\Omega} \epsilon \langle \mathbf{u}, \mathbf{v} \rangle}_{\mathbf{M}(\mathbf{u}, \mathbf{v})} \quad (3.2)$$

composed with the stiffness form  $\mathbf{K}(\mathbf{u}, \mathbf{v})$  and the mass form  $\mathbf{M}(\mathbf{u}, \mathbf{v})$ , and the linear form

$$\mathbf{L}(\mathbf{u}) = \int_{\Omega} \langle \mathbf{j}, \mathbf{v} \rangle + \int_{\partial\Omega} \langle \mathbf{g}, \mathbf{v} \rangle \quad (3.3)$$

and the appropriate Hilbert space  $\mathcal{H}_{\text{curl}}(\Omega)$  defined in (2.11). Notice that in the case of the imperfectly conducting boundary, the integral along the boundary will have to be moved from the linear to the bilinear form, since then  $\mathbf{g}$  will depend on  $\mathbf{u}$  via (2.20).

A sequence of appropriate finite dimensional spaces  $\mathcal{H}_{\text{curl},h}(\Omega)$  parametrized by  $h$  is introduced. As  $h \rightarrow 0$ , this sequence should become dense in  $\mathcal{H}_{\text{curl}}(\Omega)$ , loosely meaning that for any element in  $\mathcal{H}_{\text{curl}}(\Omega)$  we may find a sequence  $\mathbf{w}_h \in \mathcal{H}_{\text{curl},h}(\Omega)$  which approximates this element arbitrarily well in the limit. The Galerkin problem is then formulated as (see [10] for details)

Galerkin problem for the time-harmonic potential equation

$$\text{Find } \mathbf{u}_h \in \mathcal{H}_{\text{curl},h}(\Omega), \text{ such that } \mathbf{a}_\omega(\mathbf{u}_h, \mathbf{v}_h) = \mathbf{L}(\mathbf{v}_h), \forall \mathbf{v}_h \in \mathcal{H}_{\text{curl},h}(\Omega) \quad (3.4)$$

One class of finite elements, the Nédélec elements of the first kind, are particularly well suited for discretizing curl-curl-problems of the type we have derived in Section 2 (see [8]).

I refer to the FEM matrix representations in the vertex basis of the forms  $\mathbf{K}(\mathbf{u}, \mathbf{v})$  and  $\mathbf{M}(\mathbf{u}, \mathbf{v})$  distinguished in (3.2) as the stiffness matrix  $\underline{\mathbf{K}}$  and mass matrix  $\underline{\mathbf{M}}$ .

#### 3.2 NUMERICAL APPROXIMATION OF PDES USING FENICS

FEniCS<sup>1</sup> bundles a collection of Python modules designed to automate solving a Partial Differential Equation (PDE). Inspired by the demonstrations encountered in

<sup>1</sup><https://fenicsproject.org/>

[6], I will now guide you through a simple example, relevant to the context of this report to show the process of obtaining approximate solutions to PDEs with FEniCS.

Consider the time-harmonic potential equation (2.10) with the computational domain  $\Omega$  being a cubic cavity with an inlet  $\Gamma_N$  on one of its sides, but all other boundaries being perfect conductors. Set  $\mu = \epsilon = 1$  and  $\mathbf{j} = 0$  for simplicity.

The fenics package is imported along with numpy for array manipulations.

```
1 | import numpy as np
2 | import fenics as fen
```

A mesh for the cubic cavity  $\Omega$  is generated by dividing the cube into a  $10 \times 10 \times 10$  grid, whose cells are again subdivided into tetrahedrons.

```
5 | nx, ny, nz = 10, 10, 10
6 | mesh = fen.UnitCubeMesh(nx, ny, nz)
```

Our function space  $H_{curl,h}(\Omega)$  is composed using piecewise linear Nédélec elements of the first kind.

```
9 | V = fen.FunctionSpace(mesh, 'N1curl', 1)
```

The inlet is introduced at  $x = 0$ .

```
12 | class Inlet(fen.SubDomain):
13 |     def inside(self, x, on_boundary):
14 |         return on_boundary and fen.near(x[0], 0)
```

All other boundaries are perfectly conducting walls.

```
17 | class PECWalls(fen.SubDomain):
18 |     def inside(self, x, on_boundary):
19 |         return on_boundary and not Inlet().inside(x, on_boundary)
```

A mesh function is used to identify the different boundaries. It evaluates to 0, if a vertex is not on any boundary; 1 if the vertex is a the inlet; and 2 if the vertex sits on a perfectly conducting boundary.

```
22 | boundary_id = fen.MeshFunction('size_t', mesh, mesh.topology().dim()-1)
23 | boundary_id.set_all(0)
24 | Inlet().mark(boundary_id, 1)
25 | PECWalls().mark(boundary_id, 2)
```

For Nédélec elements of the first kind, (2.18) is enforced through

```
28 | u_D = fen.Expression((0.0, 0.0, 0.0), degree=2)
29 | bc = fen.DirichletBC(V, u_D, boundary_id, 2)
```

Let  $\mathbf{g} = \mathbf{e}_z$  in (2.15), which corresponds to a magnetic field  $\mu^{-1}\mathbf{B} = \mathbf{e}_y$ , and define a boundary measure.

```
32 | g = fen.Expression((0.0, 0.0, 1.0), degree=2)
33 | ds = fen.Measure('ds', subdomain_data=boundary_id)
```

Trial and test functions for the function space  $H_{curl,h}(\Omega)$  are instantiated to express the weak formulation (cf. Equation 2.13).

```
36 | u = fen.TrialFunction(V)
37 | v = fen.TestFunction(V)
```

The linear form (3.3) is assembled.

```
40 | N = fen.assemble(fen.dot(g, v) * ds(2))
```

The stiffness matrix (i.e. the first term in the bilinear form (3.2)) is assembled, and the Dirichlet boundary conditions are applied.

```
43 | K = fen.assemble(fen.dot(fen.curl(u), fen.curl(v)) * fen.dx)
44 | bc.apply(K)
```

The mass matrix (i.e. the second term in the bilinear form (3.2)) is assembled. The fact that the Dirichlet boundary conditions have already been separately applied to the stiffness matrix above is accounted for by setting all rows and columns corresponding to degrees of freedom on the Dirichlet boundary to zero.

```
47 | M = fen.assemble(fen.dot(u, v) * fen.dx)
48 | bc.zero(M)
```

A function to compute an approximation of the  $L_2(\Omega)$ -norm of a solution to the system can be created. Notice how I can reuse the mass matrix  $\underline{M}$  for this purpose only due to the fact that  $\epsilon = 1$  was taken.

```
51 | def L2_norm(u):
52 |     u_vec = u.vector().get_local()
53 |     return pow(((M * u_vec) * u_vec).sum(), 0.5)
```

Finally, for 200 uniformly spaced frequencies  $\omega \in [6.2, 6.8]$ , the approximate solution to the cubic cavity at each of these frequencies is computed and its  $L_2(\Omega)$ -norm memorized for later.

```
56 | omegas = np.linspace(6.2, 6.8, 200)
57 | norms = []
58 | u = fen.Function(V)
59 | for omega in omegas:
60 |     fen.solve(K - omega**2 * M, u.vector(), N)
61 |     norms.append(L2_norm(u))
```

What results is an approximation of the frequency response in the  $L_2(\Omega)$ -norm for the cubic cavity (see Figure 3.1).

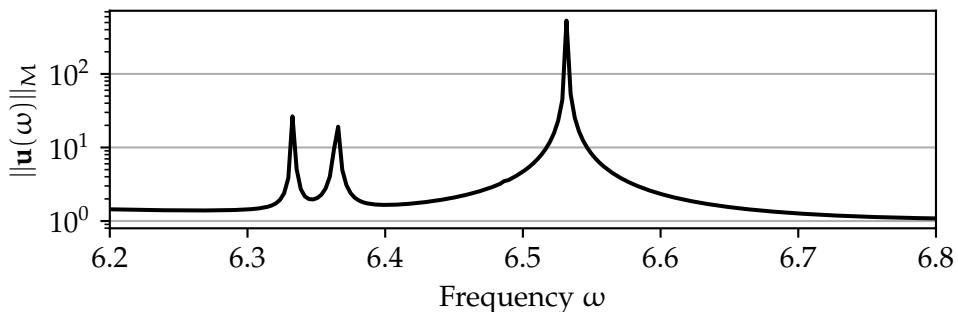


FIGURE 3.1 – Approximate frequency response in the  $L_2(\Omega)$ -norm of a cubic cavity with one face acting as an inlet and all others as perfectly conducting boundaries. At resonant frequencies, the  $L_2(\Omega)$ -norm theoretically tends to infinity. Numerically, they appear as finite peaks in the frequency response.

## 4 MINIMAL RATIONAL INTERPOLATION FOR THE TIME-HARMONIC MAXWELL'S EQUATIONS

Let  $\mathbf{u} : \mathbb{C} \ni \omega \mapsto H_{\text{curl}}$ . Given “snapshots” of the vector field  $\mathbf{u}(\omega_j)$  at  $\omega_j$  for  $j \in \{1, \dots, S\}$ , the goal is to find a surrogate  $\tilde{\mathbf{u}}$  that locally (i.e. near  $\omega_1, \dots, \omega_S$ ) satisfies

$$\tilde{\mathbf{u}}(\omega) \approx \mathbf{u}(\omega) \quad (4.1)$$

This may be achieved using the Minimal Rational Interpolation (MRI) method, which I will motivate, discuss, and extend in the following.

### 4.1 MOTIVATION

In the most simple case (dropping all constants), equations of the type (2.10) take the form

$$\nabla \times (\nabla \times \mathbf{u}) - \omega^2 \mathbf{u} = \mathbf{j} \quad (4.2)$$

Transitioning to a finite element space and expressing (4.2) in terms of the stiffness matrix  $\underline{\mathbf{K}}$  and mass matrix  $\underline{\mathbf{M}}$  (see Section 3.1) allows for us to write

$$(\underline{\mathbf{K}} - \omega^2 \underline{\mathbf{M}}) \mathbf{u} = \mathbf{j} \implies \mathbf{u} = (\underline{\mathbf{K}} - \omega^2 \underline{\mathbf{M}})^{-1} \mathbf{j} \quad (4.3)$$

An eigenvalue decomposition  $\underline{\mathbf{K}} = \underline{\mathbf{V}} \underline{\Lambda} \underline{\mathbf{V}}^H$  of the hermitian stiffness matrix  $\underline{\mathbf{K}}$  leads to a similar form of the solution as the one seen in [1]

$$\mathbf{u} = \underline{\mathbf{V}} (\underline{\Lambda} - \omega^2 \underline{\mathbf{1}})^{-1} \underline{\mathbf{V}}^H \mathbf{j} = \sum_i \frac{\mathbf{v}_i \langle \mathbf{v}_i, \mathbf{j} \rangle}{\lambda_i - \omega^2} \quad (4.4)$$

This follows from the fact that  $\underline{\Lambda}$  is diagonal, hence also  $(\underline{\Lambda} - \omega^2 \underline{\mathbf{1}})^{-1}$ . Here, the diagonal elements of  $\underline{\Lambda}$  are denoted with  $\lambda_i$  (the eigenvalues of  $\underline{\mathbf{A}}$ ) and the columns of  $\underline{\mathbf{V}}$  with  $\mathbf{v}_i$  (the eigenvectors of  $\underline{\mathbf{A}}$ ).

With the expression of the solution  $\mathbf{u}$  in terms of a rational polynomial function (see (4.4)), we can motivate why rational interpolation is a valid approach for approximating  $\mathbf{u}$ . Some alternatives such as polynomial interpolation are not as capable to model the singularities at the resonant frequencies  $\omega^2 = \lambda_i$ .

Subsequently, the goal is to find rational surrogates of the form

$$\tilde{\mathbf{u}}(\omega) = \frac{\mathbf{P}(\omega)}{\mathbf{Q}(\omega)} \quad (4.5)$$

with

$$\mathbf{P}(\omega) = \sum_i \frac{\mathbf{p}_i}{\omega - \omega_i} \quad (4.6)$$

and

$$\mathbf{Q}(\omega) = \sum_i \frac{\mathbf{q}_i}{\omega - \omega_i} \quad (4.7)$$

in the barycentric representation.

## 4.2 MINIMAL RATIONAL INTERPOLATION

In the following, I denote with

$$\langle \bar{\mathbf{u}}, \bar{\mathbf{v}} \rangle_M = \bar{\mathbf{u}}^H \underline{\mathbf{M}} \bar{\mathbf{v}} \approx \int_{\Omega} \langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle_{L_2(\Omega)} \quad (4.8)$$

the finite element approximation of the inner product in  $L_2(\Omega)$ . With  $\bar{\mathbf{u}}, \bar{\mathbf{v}} \in \mathbb{C}^N$  I denote the vertex basis representations of the vector fields  $\mathbf{u}, \mathbf{v}$ , i.e. the collection of all  $N$  Degrees of Freedom (DOF) in a vector, while  $\underline{\mathbf{M}}$  is the matrix representing the inner product in this vertex basis. Similarly, let

$$\|\bar{\mathbf{u}}\|_M = \sqrt{\langle \bar{\mathbf{u}}, \bar{\mathbf{u}} \rangle_M} \approx \|\mathbf{u}\|_{L_2(\Omega)} \quad (4.9)$$

Because from here on, exclusively the vertex basis representation  $\bar{\mathbf{u}}$  will be used, I will not go through the trouble to put a bar on each such variable but will instead also refer to it with  $\mathbf{u}$ .

For completeness, I state the strategy for numerically computing the MRI for a collection of snapshots sampled from the target  $\mathbf{u}$  in Algorithm 1 [2]. The heart of the algorithm consists in computing the Singular Value Decomposition (SVD) of a Gramian matrix  $\underline{\mathbf{G}}$ , and using the last left-singular vector to build the surrogate.

---

### Algorithm 1 Minimal rational interpolation

---

**Require:**  $\omega_1, \dots, \omega_S$  ▷ Support points  
**Require:**  $\mathbf{u}(\omega_1), \dots, \mathbf{u}(\omega_S)$  ▷ Snapshots  
 Compute  $\underline{\mathbf{G}}$  with  $g_{ij} = \langle \mathbf{u}(\omega_i), \mathbf{u}(\omega_j) \rangle_M$ ,  $i, j \in \{1, \dots, S\}$  ▷ Gramian matrix  
 Compute the singular value decomposition  $\underline{\mathbf{G}} = \underline{\mathbf{V}} \Sigma \underline{\mathbf{V}}^H$   
 Define  $\mathbf{q} = (q_1, \dots, q_S)^T = \underline{\mathbf{V}}[:, S]$   
 Define  $\tilde{\mathbf{u}}(\omega) = \mathbf{P}(\omega)/Q(\omega)$  with  $\mathbf{P}(\omega) = \sum_{j=1}^S \frac{q_j \mathbf{u}(\omega_j)}{\omega - \omega_j}$  and  $Q(\omega) = \sum_{j=1}^S \frac{q_j}{\omega - \omega_j}$

---

Because the Gramian matrix  $\underline{\mathbf{G}}$  is hermitian, the singular value decomposition is the eigenvalue decomposition. In the algorithm, I implicitly assume that the singular values of  $\underline{\mathbf{G}}$  are ordered in decreasing order, meaning that  $\underline{\mathbf{V}}[:, S]$  is the left singular vector corresponding to the smallest singular value. In most numerical packages, such as the python package numpy, this is automatically the case.

## 4.3 GREEDY MINIMAL RATIONAL INTERPOLATION

A question that arises from the previous section is what the ideal choice of support points  $\omega_1, \dots, \omega_S$  is: How many support points are required to achieve a good enough approximation, and how should the support points be distributed in an interval on which we want to compute the approximation? The Greedy Minimal Rational Interpolation (gMRI) algorithm [13] tackles both questions simultaneously by sequentially adding a new support points to be included in the build of the minimal rational surrogate.

In brief, the algorithm starts with a set of candidate support points  $\Omega_{\text{test}} = \{\omega_i\}_{i=1}^M$ , for which we can guarantee to find an approximate solution to (3.4). From

the  $\Omega_{\text{test}}$  a subset is chosen (usually the smallest and largest element), and (3.4) is solved for these two initial supports. Using these solutions, the rational surrogate  $P(\omega)/Q(\omega)$  is built with MRI (Algorithm 1). Motivated by the expression for the upper bound on the residual norm demonstrated in [11], new support points are chosen as the minimizers of the previously computed denominator polynomial  $Q(\omega)$  and added to the existing set of support points. This procedure is iteratively continued until the relative error norm drops below a certain tolerance set in advance.

The gMRI algorithm can be found in Algorithm 2.

---

**Algorithm 2** Greedy minimal rational interpolation

---

```

Require:  $\tau > 0$                                  $\triangleright$  Relative  $L_2$ -error tolerance
Require:  $\Omega_{\text{test}} = \{\omega_i\}_{i=1}^M$        $\triangleright$  Set of candidate support points
Require:  $a_\omega(\mathbf{u}, \mathbf{v}) = L(\mathbf{v})$            $\triangleright$  Finite element formulation of the problem

Initialize  $t \geq 1$ 
Choose  $\omega_1, \dots, \omega_t \in \Omega_{\text{test}}$             $\triangleright$  Usually the smallest and largest element
Remove  $\omega_1, \dots, \omega_t$  from  $\Omega_{\text{test}}$ 
Solve  $a_{\omega_i}(\mathbf{u}_i, \mathbf{v}) = L(\mathbf{v})$  for  $i \in \{1, \dots, t\}$ 
Build surrogate  $\tilde{\mathbf{u}}_t = P_t(\omega)/Q_t(\omega)$  using the solutions  $\mathbf{u}_1, \dots, \mathbf{u}_t$ 
while  $\Omega_{\text{test}} \neq \emptyset$  do
     $\omega_{t+1} \leftarrow \operatorname{argmin}_{\omega \in \Omega_{\text{test}}} |Q_t(\omega)|$ 
    Remove  $\omega_{t+1}$  from  $\Omega_{\text{test}}$ 
    Solve  $a_{\omega_{t+1}}(\mathbf{u}_{t+1}, \mathbf{v}) = L(\mathbf{v})$ 
    Build surrogate  $\tilde{\mathbf{u}}_{t+1} = P_{t+1}(\omega)/Q_{t+1}(\omega)$  with MRI of  $\mathbf{u}_1, \dots, \mathbf{u}_{t+1}$  (Algorithm 1)
    if  $\|\mathbf{u}_{t+1} - \tilde{\mathbf{u}}_t(\omega_{t+1})\|_M / \|\mathbf{u}_{t+1}\|_M < \tau$  then return
    end if
     $t \leftarrow t + 1$ 
end while

```

---

#### 4.4 PROPERTIES OF THE RATIONAL SURROGATE

Multiplying the numerator and denominator of the rational surrogate  $\tilde{\mathbf{u}}$  (4.5) with the nodal polynomial associated with the support points yields

$$\tilde{\mathbf{u}}(\omega) = \sum_{j=1}^s \prod_{\substack{i=1 \\ i \neq j}}^s (\omega - \omega_i) q_j \mathbf{u}(\omega_j) / \sum_{j=1}^s \prod_{\substack{i=1 \\ i \neq j}}^s (\omega - \omega_i) q_j \quad (4.10)$$

Hence, if the rational surrogate  $\tilde{\mathbf{u}}$  is evaluated at one of the interpolation nodes  $\omega_i$ , the snapshot  $\mathbf{u}(\omega_i)$  supplied to the MRI algorithm is recovered. This shows that the rational surrogate indeed satisfies the interpolation property.

#### 4.5 FINDING ROOTS OF RATIONAL FUNCTIONS

If the rational surrogate  $\tilde{\mathbf{u}}$  is evaluated in a zero  $\omega^*$  of the denominator  $Q(\omega^*) = 0$ , we observe a pole, provided  $P(\omega^*)$  does not also vanish in that frequency.  $\omega^*$  is referred to as a resonant frequency.

In order to find the approximate resonant frequencies of a system, we simply need to perform the following steps:

1. Compute the rational surrogate using MRI or gMRI.
2. Determine the zeros of the denominator  $Q(\omega)$  of the surrogate

The first step was already elaborated upon in Sections 4.2 and 4.3. Finding the zeros of the denominator of the minimal rational surrogate, a rational function of the form (4.7) can be elegantly converted to an eigenvalue problem [5]:

Define

$$v_i(\omega) = (\omega - \omega_i)^{-1} \quad (4.11)$$

We want to find  $\omega$ , such that

$$0 = Q(\omega) = \sum_{i=1}^S q_i v_i(\omega) \quad (4.12)$$

This can be equivalently expressed as the generalized eigenvalue problem

$$\underline{\mathbf{A}}\mathbf{w} = \omega \underline{\mathbf{B}}\mathbf{w} \quad (4.13)$$

with

$$\underline{\mathbf{A}} = \begin{pmatrix} 0 & q_1 & q_2 & \dots & q_S \\ 1 & \omega_1 & & & \\ 1 & & \omega_2 & & \\ \vdots & & & \ddots & \\ 1 & & & & \omega_S \end{pmatrix} \text{ and } \underline{\mathbf{B}} = \begin{pmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \quad (4.14)$$

Solving the eigenproblem (4.13) numerically will confidently yield two spurious, infinite eigenvalues which, however, can be filtered out manually with ease. The eigenvector corresponding to the eigenvalue  $\omega$  is  $\mathbf{w}(\omega) = (1, v_1(\omega), \dots, v_S(\omega))^T$  [5]. Consequently, each eigenvalue  $\omega$  on one hand satisfies the constraints (4.11) and with (4.12) is also the zero that we set out to find.

## 4.6 OPTIMIZATION TRICKS FOR GREEDY MINIMAL RATIONAL INTERPOLATION

There exist many ways of improving the efficiency and/or capability of the gMRI algorithm [12]. In the following, I will present a small collection of them.

### 4.6.1 ADDITIVE HOUSEHOLDER DECOMPOSITION

Algorithm 1 requires the computation of the Singular Value Decomposition (SVD) of the Gramian matrix  $\underline{\mathbf{G}}$  in order to build the rational surrogate. A more efficient and better conditioned [12] alternative is to instead compute the QR decomposition of the snapshot matrix  $\underline{\mathbf{U}} = [\mathbf{u}(\omega_1), \dots, \mathbf{u}(\omega_S)]$ . Since

$$\underline{\mathbf{G}} = \underline{\mathbf{U}}^H \underline{\mathbf{M}} \underline{\mathbf{U}} \quad (4.15)$$

with the matrix  $\underline{\mathbf{M}}$  representing the finite element inner product in  $L_2(\Omega)$  in the basis of the mesh vertices. A QR decomposition with respect to the inner product  $\langle \mathbf{u}, \mathbf{v} \rangle_M = \mathbf{u}^H \underline{\mathbf{M}} \mathbf{v}$  yields  $\underline{\mathbf{U}} = \underline{\mathbf{Q}} \underline{\mathbf{R}}$  with  $\underline{\mathbf{Q}}^H \underline{\mathbf{M}} \underline{\mathbf{Q}} = \underline{\mathbf{I}}$ . When plugging this into (4.15) one sees

$$\underline{\mathbf{G}} = (\underline{\mathbf{Q}} \underline{\mathbf{R}})^H \underline{\mathbf{M}} (\underline{\mathbf{Q}} \underline{\mathbf{R}}) = \underline{\mathbf{R}}^H \underline{\mathbf{R}} \quad (4.16)$$

Let the SVD of  $\underline{\mathbf{R}}$  be

$$\underline{\mathbf{R}} = \underline{\mathbf{W}} \underline{\mathbf{S}} \underline{\mathbf{V}}^H \quad (4.17)$$

Inserting this into (4.16) results in

$$\underline{\mathbf{G}} = (\underline{\mathbf{W}} \underline{\mathbf{S}} \underline{\mathbf{V}}^H)^H \underline{\mathbf{W}} \underline{\mathbf{S}} \underline{\mathbf{V}}^H = \underline{\mathbf{V}} \underline{\mathbf{S}}^2 \underline{\mathbf{V}}^H \quad (4.18)$$

which coincides with the SVD of the Gramian matrix  $\underline{\mathbf{G}}$  if we take the square root of the singular values. Since for MRI only really the last left singular vector in  $\underline{\mathbf{V}}$  is of interest, computing  $\underline{\mathbf{G}}$  explicitly may be avoided in this way.

There is one additional benefit to taking the route via the SVD of  $\underline{\mathbf{R}}$  instead of  $\underline{\mathbf{G}}$  for building the surrogate: When extending the snapshot matrix by an additional snapshot  $\mathbf{u}(\omega_{S+1})$ , the resulting triangular matrix  $\underline{\mathbf{R}}^{(S+1)}$  from a QR decomposition on this extended snapshot matrix only differs from the (usually already computed) matrix  $\underline{\mathbf{R}}^{(S)}$  only in the last column. Thus, it is possible to reuse many results obtained in a previous iteration during the gMRI algorithm and therefore significantly increase computational efficiency.

I developed such an additive QR decomposition in Algorithm 3, which results from an adaption of the Householder triangularization algorithm found in [14]. In essence, this algorithm takes the triangular matrix  $\underline{\mathbf{R}}$ , orthonormal matrix  $\underline{\mathbf{E}}$ , and Householder matrix  $\underline{\mathbf{V}}$  and extends each of them according to the additional snapshots supplied to the algorithm.

Notice that towards the end of Algorithm 3 I have introduced the shorthand

$$\langle \underline{\mathbf{E}}[S+1:j,:], \underline{\mathbf{V}}[j,:]\rangle_M \underline{\mathbf{E}}[S+1:j,:] \quad (4.19)$$

for

$$\sum_{i=S+1}^{j-1} \langle \underline{\mathbf{E}}[i,:], \underline{\mathbf{V}}[j,:]\rangle_M \underline{\mathbf{E}}[i,:] \quad (4.20)$$

motivated by the performance boost an array multiplication with the Python package numpy introduces, compared to a simple for loop for summing.

#### 4.6.2 STABILITY OF SINGULAR VALUE DECOMPOSITION

The stability of Algorithm 1 used to build the rational surrogate using MRI can be checked by analyzing the singular values  $\sigma_1, \dots, \sigma_S$  obtained from performing the SVD of  $\underline{\mathbf{G}}$ . Assume these values to be ordered in descending order, which the Python package numpy automatically does when computing the SVD of a matrix. The conditioning of the problem may be measured with the relative spectral gap [12]

$$\frac{\sigma_{S-1} - \sigma_S}{\sigma_1 - \sigma_S} \quad (4.21)$$

---

**Algorithm 3** Additive Householder triangularization

---

**Require:**  $\underline{U}[1 \dots s, 1 \dots N]$  ▷ Next snapshot matrix  
**Require:**  $\underline{R}[1 \dots S, 1 \dots S]$  ▷ Previous triangular matrix  
**Require:**  $\underline{E}[1 \dots S, 1 \dots N]$  ▷ Previous orthonormal matrix  
**Require:**  $\underline{V}[1 \dots S, 1 \dots N]$  ▷ Previous Householder matrix

Extend size of  $\underline{R}$  to  $(S + s) \times (S + s)$   
Extend  $\underline{E}$  with  $S$  orthonormal columns to  $(S + s) \times N$   
Extend size of  $\underline{V}$  to  $(S + s) \times N$

**for**  $j = S + 1 : S + s$  **do**

- $\mathbf{u} = \underline{U}[j, :]$
- for**  $k = 1 : j - 1$  **do**
- $\mathbf{u} \leftarrow \mathbf{u} - 2\langle \underline{V}[k, :], \mathbf{u} \rangle_M \underline{V}[k, :]$
- $\underline{R}[k, j] \leftarrow \langle \underline{E}[k, :], \mathbf{u} \rangle_M$
- $\mathbf{u} \leftarrow \mathbf{u} - \underline{R}[k, j] \underline{E}[k, :]$
- end for**
- $\underline{R}[j, j] \leftarrow \|\mathbf{u}\|_M$
- $\alpha \leftarrow \langle \underline{E}[j, :], \mathbf{u} \rangle_M$
- if**  $|\alpha| \neq 0$  **then**
- $\underline{E}[j, :] \leftarrow \underline{E}[j, :](-\alpha/|\alpha|)$
- end if**
- $\underline{V}[j, :] \leftarrow \underline{R}[j, j] \underline{E}[j, :] - \mathbf{u}$
- $\underline{V}[j, :] \leftarrow \underline{V}[j, :] - \langle \underline{E}[S + 1 : j, :], \underline{V}[j, :] \rangle_M \underline{E}[S + 1 : j, :]$  ▷ numpy shorthand
- $\sigma \leftarrow \|\underline{V}[j, :]\|_M$
- if**  $\sigma \neq 0$  **then**
- $\underline{V}[j, :] \leftarrow \underline{V}[j, :]/\sigma$
- else**
- $\underline{V}[j, :] \leftarrow \underline{E}[j, :]$
- end if**

**end for**

---

Usually, values of approximately above  $10^{13}$  indicate that the rational surrogate could not be built in a stable way and in my implementations the user is informed by a warning.

#### 4.6.3 ALTERNATIVE REPRESENTATIONS OF THE SURROGATE

Denote with  $\underline{\mathbf{U}} = [\mathbf{u}(\omega_1), \dots, \mathbf{u}(\omega_S)]$  the snapshot matrix. Let

$$\hat{\mathbf{u}}(\omega) = \sum_{j=1}^S \frac{q_j \mathbf{e}_j}{\omega - \omega_j} / \sum_{j=1}^S \frac{q_j}{\omega - \omega_j} \quad (4.22)$$

with the canonical basis vectors  $\{\mathbf{e}_j\}_j$ , and denote  $\hat{\underline{\mathbf{U}}} = [\hat{\mathbf{u}}(\omega_1), \dots, \hat{\mathbf{u}}(\omega_S)]$ . Inspecting the rational surrogate (defined in Algorithm 1) closely, one can see that the rational surrogate can be recovered from  $\hat{\mathbf{u}}$  via

$$\tilde{\mathbf{u}}(\omega) = \underline{\mathbf{U}} \hat{\mathbf{u}}(\omega) \quad (4.23)$$

Therefore, provided we know the snapshot matrix  $\underline{\mathbf{U}}$ , a rational surrogate is fully characterized by just  $S$  numbers  $\{q_1, \dots, q_S\}$  and the locations of the interpolation nodes  $\{\omega_1, \dots, \omega_S\}$ .

Additionally, let

$$\hat{\mathbf{u}}(\omega) = \sum_{j=1}^S \frac{q_j \mathbf{r}_j}{\omega - \omega_j} / \sum_{j=1}^S \frac{q_j}{\omega - \omega_j} \quad (4.24)$$

with  $\mathbf{r}_j$  being the  $j$ -th column in the triangular matrix  $\underline{\mathbf{R}}$  stemming from the QR decomposition of the snapshot matrix  $\underline{\mathbf{U}} = \underline{\mathbf{Q}} \underline{\mathbf{R}}$ . Then, the original rational surrogate can again be recovered via

$$\tilde{\mathbf{u}}(\omega) = \underline{\mathbf{Q}} \hat{\mathbf{u}}(\omega) \quad (4.25)$$

$\hat{\mathbf{u}}(\omega)$  provides us with a simplified alternative to the computation of the relative error which is used as a stopping criterion in gMRI (Algorithm 2):

First, one may use the  $\underline{\mathbf{M}}$ -orthonormality of  $\underline{\mathbf{Q}}$  to get

$$\|\mathbf{u}_{t+1}\|_{\mathcal{M}} = \|\underline{\mathbf{Q}}^{(t+1)} \mathbf{r}_{t+1}\|_{\mathcal{M}} = \|\mathbf{r}_{t+1}\| \quad (4.26)$$

Where  $\|\cdot\|$  is the Euclidean norm. Similarly, it follows that

$$\begin{aligned} \|\mathbf{u}_{t+1} - \tilde{\mathbf{u}}_t(\omega_{t+1})\|_{\mathcal{M}} &= \|\underline{\mathbf{Q}}^{(t+1)} \mathbf{r}_{t+1} - \underline{\mathbf{Q}}^{(t+1)} \hat{\mathbf{u}}'_t(\omega_{t+1})\|_{\mathcal{M}} \\ &= \|\mathbf{r}_{t+1} - \hat{\mathbf{u}}'_t(\omega_{t+1})\| \end{aligned} \quad (4.27)$$

with the short hand  $\hat{\mathbf{u}}'_t = [\hat{\mathbf{u}}_t^\top, 0]^\top$  and where  $\mathbf{r}_{t+1}$  is the last column in  $\underline{\mathbf{R}}^{(t+1)}$ , the triangular matrix from the  $(t+1)$ -th step in the additive Householder decomposition (Algorithm 3).

In this way, the particular relative error norm required in Algorithm 2 can be computed with only Euclidean norms of relatively low-dimensional vectors:

$$\frac{\|\mathbf{u}_{t+1} - \tilde{\mathbf{u}}_t(\omega_{t+1})\|_{\mathcal{M}}}{\|\mathbf{u}_{t+1}\|_{\mathcal{M}}} = \frac{\|\mathbf{r}_{t+1} - \hat{\mathbf{u}}'_t(\omega_{t+1})\|}{\|\mathbf{r}_{t+1}\|} \quad (4.28)$$

Notice how on the left-hand side, the vectors of dimension  $N$  (usually with  $N$  of the order  $10^4$ ) are multiplied with an  $N \times N$  matrix to compute the norm, whereas the vectors on the right-hand side are only of size  $S$  (usually  $\leq 20$ ) and do not require a matrix for the norm computation.

## 5 EXAMPLES

The goal is now to apply the techniques introduced in the previous sections to three examples (see Figure 2.1 to get an idea about the difference and similarities between these examples).

### 5.1 TWO-DIMENSIONAL RECTANGULAR RESONANT CAVITY

Based on the theoretical considerations from Section 5.1, a rectangular resonant cavity with dimensions  $L_x = 5$  and  $L_y = 1$  (see Figure 5.1) is studied.

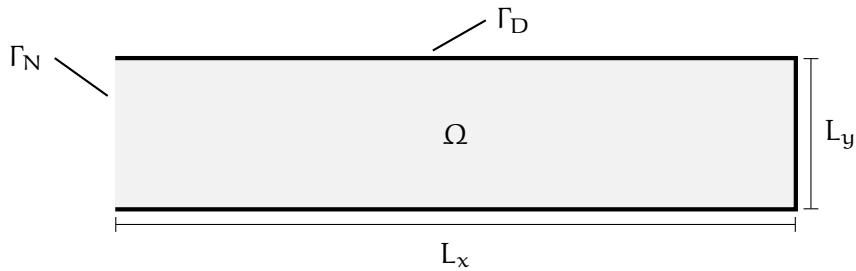


FIGURE 5.1 – The rectangular resonant cavity is a medium  $\Omega$  enclosed by a perfectly conducting boundary  $\Gamma_D$  and an inlet  $\Gamma_N$  chosen to coincide with the rectangle's edge at  $x = 0$  for the experiments in this section.

For simplicity, I set  $\epsilon = \mu = 1$ . A uniform grid with 101 subdivisions in the  $x$ - and 21 in the  $y$ -direction whose cells are again subdivided by their diagonals was used to generate a highly symmetric triangular mesh that allows for 4365 degrees of freedom. The system is forced from the inlet at  $x = 0$  with  $g_z(y) = \sin(\pi y/L_y)$ .

#### 5.1.1 EXPLORATION OF THE PROBLEM

To give the reader an impression of what the solution  $\mathbf{u}_z$  to this problem looks like, I plot the FEM-solution at the first and sixth numerical resonant frequencies in Figures 5.2 and 5.3 respectively. Notice that I type its variable in bold, since it is a large vector with the values for each of the DOFs as its components.

I now study the gMRI for the quantity  $u_z$ . Algorithm 2 is performed with a tolerance of  $\tau = 10^{-2}$ . The candidate support points  $\Omega_{\text{test}}$  are 1000 uniformly spaced points in  $\omega = [3, 5]$ . The FEM solution  $\mathbf{u}_z$  is computed and the gMRI surrogate  $\tilde{\mathbf{u}}_z$  evaluated at all points in  $\Omega_{\text{test}}$ , and the  $\|\cdot\|_M$ -norm for both is plotted in Figure 5.4.

For the same problem, the approximation of the relative error in the  $\|\cdot\|_M$ -norm of the rational surrogate  $\tilde{\mathbf{u}}_z^{(S)}$  from the FEM solution in every second iteration  $S$  of the gMRI algorithm is shown in Figure 5.5. This time, however, the candidate support points for the rational surrogate  $\tilde{\mathbf{u}}_z^{(S)}$  and the points at which the FEM-solution and the rational surrogate are compared are completely out of sync. Because if they coincided, the relative error would obviously tend to zero at each of the support points chosen for building the surrogate, which would obscure the point I want to make with this plot.

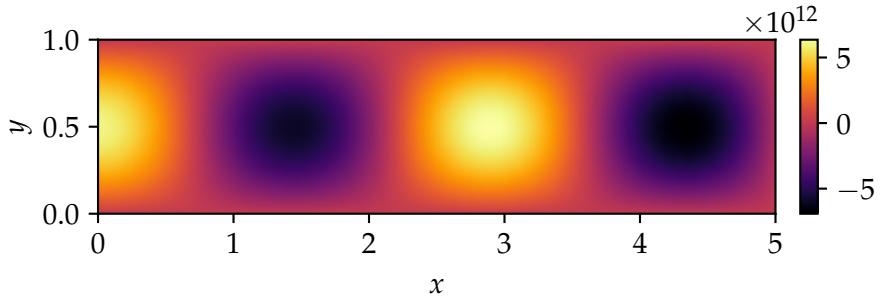


FIGURE 5.2 – The solution  $\mathbf{u}_z$  obtained with the FEM at the first numerical resonant frequency of the cavity. Observe how at every perfectly conducting boundary  $\Gamma_D$  (cf. Figure 5.1) the solution gradually goes to zero, as I imposed.

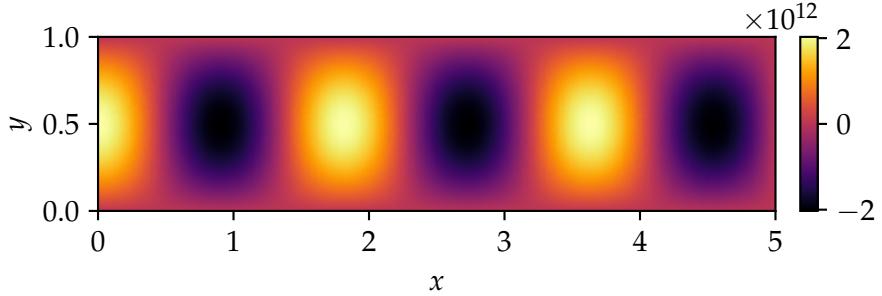


FIGURE 5.3 – The solution  $\mathbf{u}_z$  obtained with the FEM at the sixth numerical resonant frequency of the cavity.

### 5.1.2 NUMERICAL APPROXIMATION OF RESONANT FREQUENCIES

As already discussed in Section 4.5, gMRI may be used to approximate resonant frequencies of a system. Conveniently enough, the analytical resonant frequencies for a rectangular resonant cavity as the one currently being studied are easy to determine due to the homogeneous boundary conditions:

$$\omega_{n,m} = \pi \sqrt{\left(\frac{2n+1}{2L_x}\right)^2 + \left(\frac{m}{L_y}\right)^2}, \quad n \in \{0, 1, \dots\}, \quad m \in \{1, 2, \dots\} \quad (5.1)$$

Conventionally, resonant frequencies can be determined by solving the generalized (hermitian) eigenvalue problem

$$\underline{\mathbf{K}}\mathbf{w} = \omega^2 \underline{\mathbf{M}}\mathbf{w} \quad (5.2)$$

involving a stiffness matrix  $\underline{\mathbf{K}}$  and a mass matrix  $\underline{\mathbf{M}}$  (here, the ones defined in Section 3 will be used, such that the eigenvectors  $\mathbf{w}$  correspond to the solutions  $\mathbf{u}_z$  at the corresponding resonant frequencies). This can for instance be accomplished with the `eigsh` sparse eigenvalue solver included in the `scipy` Python library. However, it is first necessary to remove all the columns and rows from  $\underline{\mathbf{K}}$  and  $\underline{\mathbf{M}}$  that correspond

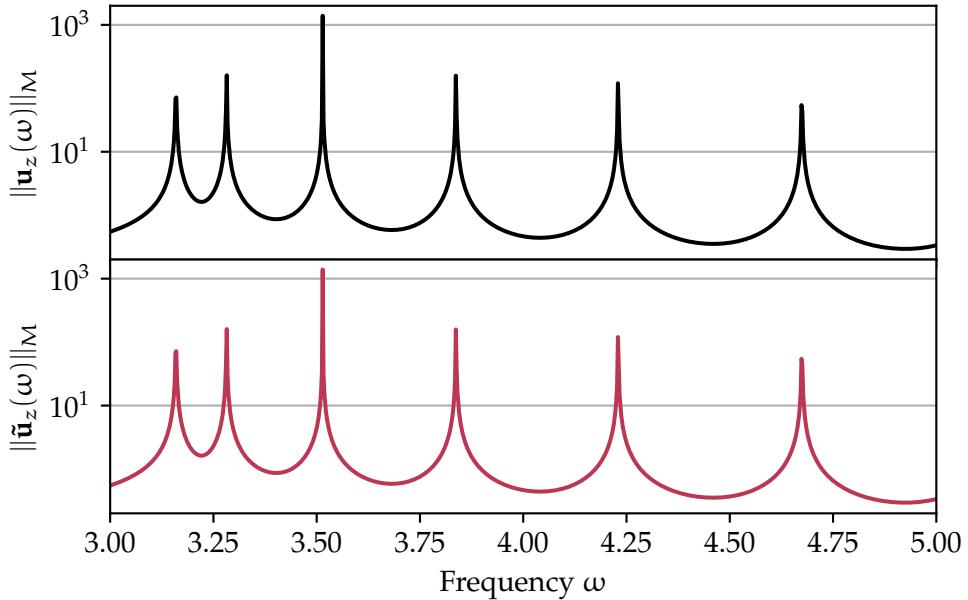


FIGURE 5.4 –  $\|\cdot\|_M$ -norm of the FEM solution  $\mathbf{u}_z$  (top) and the gMRI surrogate  $\tilde{\mathbf{u}}_z$  (bottom).

to points on the perfectly conducting boundary, since they are all zero after having assembled the matrices. This was demonstrated in Section 3.2. In Table 5.1, I compare the two approaches (eigsh and gMRI) in their mean absolute deviation  $\bar{\Delta}$  of the six resonant frequencies inside the interval  $\omega = [3, 5]$  from their corresponding analytical resonances. Furthermore, the time each method requires to produce a result given the assembled matrices  $\mathbf{K}$  and  $\mathbf{M}$  is measured over multiple runs on a 7th generation Intel 8750H CPU at 2.20 GHz using the `timeit` library. This is done for four refinements of the mesh, each resulting in 10 times more DOFs than the previous one.

TABLE 5.1 – In the same context as on the previous pages, two approaches to finding resonant frequencies: The sparse hermitian eigenvalue solver `eigsh` from the `scipy` library and the procedure using gMRI that was introduced in Section 4.5. For various DOFs the mean absolute deviation  $\bar{\Delta}$  from the analytical resonant frequencies located in the interval  $\omega \in [3, 5]$  and the time spent in computation  $t$  are shown.

	eigsh		gMRI	
DOF	$\bar{\Delta}$	$t$	$\bar{\Delta}$	$t$
713	$1.950 \times 10^{-2}$	$25.9 \pm 1.1$ ms	$1.950 \times 10^{-2}$	$61.9 \pm 3.6$ ms
7412	$1.826 \times 10^{-3}$	$199.0 \pm 9.9$ ms	$1.827 \times 10^{-3}$	$410.0 \pm 16.8$ ms
74722	$1.817 \times 10^{-4}$	$3.5 \pm 0.1$ s	$1.820 \times 10^{-4}$	$5.2 \pm 0.2$ s
745513	$1.811 \times 10^{-5}$	$75.0 \pm 1.6$ s	$1.846 \times 10^{-5}$	$104.0 \pm 1.1$ s

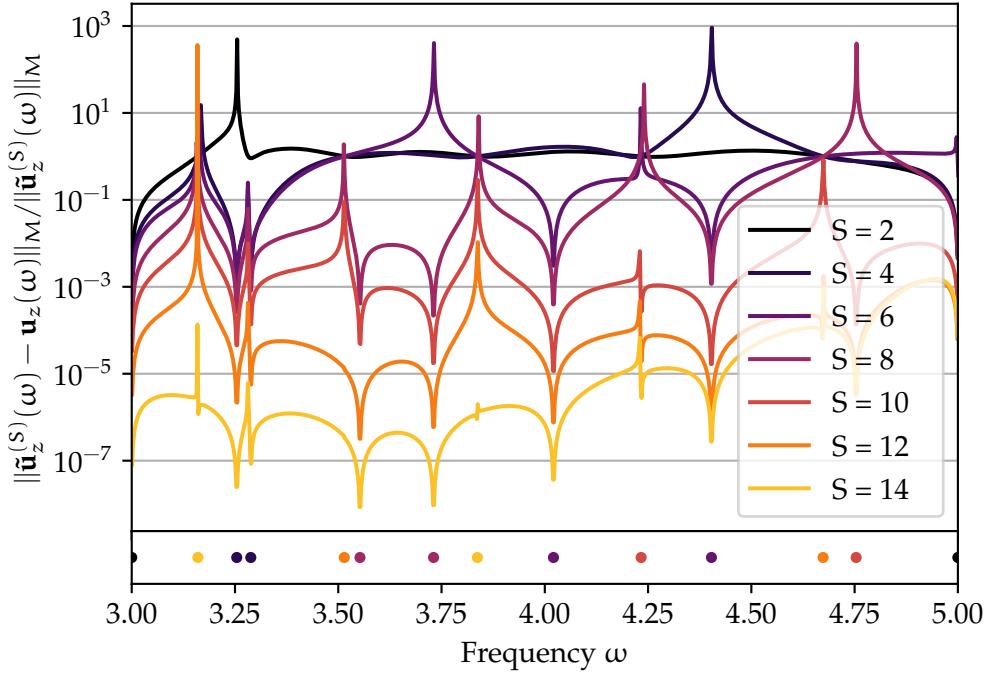


FIGURE 5.5 – Progression of a relative approximation error of the rational surrogates  $\tilde{u}_z^{(S)}$  that were built using  $S$  support points. The freshly added support points during the previous two iterations of gMRI are shown in their corresponding color at the bottom. Notice how the decrease in relative error is not uniform and wherever a support point is added, the error is locally decreased.

Hereafter I would like to discuss two major benefits the gMRI method for finding resonant modes (explained in Section 4.5) holds over the eigsh approach.

Fistly, eigsh requires us to specify the exact number of eigenvalues that should be approximated. Unless it is a priori known how many resonant modes are expected to fall within an interval of interest, there is no guarantee of having found all of them. On the other hand, the gMRI way will reliably find all resonant modes within and even in a close neighborhood of the interval of interest.

Secondly, while solving the generalized eigenvalue problem some resonant modes are incorrectly identified as such: Depending on the boundary condition at the inlet, some analytically admissible resonant modes are suppressed and do not emerge when solving the actual problem. A sketch of the reason for this caveat is demonstrated in the following.

Take  $\{\mathbf{w}_j, \omega_j^2\}_j$  to be resonant modes, i.e. solutions to the eigenvalue problem (5.2), such that

$$\underline{\mathbf{K}}\mathbf{w}_j = \omega_j^2 \underline{\mathbf{M}}\mathbf{w}_j \quad (5.3)$$

Adding a source term  $\mathbf{f}$ , the new system is described by (see Equation 2.13)

$$\underline{\mathbf{K}}\mathbf{w} - \omega^2 \underline{\mathbf{M}}\mathbf{w} = \mathbf{f} \quad (5.4)$$

If a solution  $\mathbf{w}$  can be expressed in terms of the basis  $\{\mathbf{w}_j\}_j$ , meaning  $\mathbf{w} = \sum_j \alpha_j \mathbf{w}_j$

for some  $\alpha_j$ , then (5.4) becomes

$$\sum_j \alpha_j (\underline{\mathbf{K}}\mathbf{w}_j - \omega^2 \underline{\mathbf{M}}\mathbf{w}_j) = \mathbf{f} \quad (5.5)$$

Using (5.3) yields

$$\sum_j \alpha_j (\omega_j^2 - \omega^2) \underline{\mathbf{M}}\mathbf{w}_j = \mathbf{f} \quad (5.6)$$

from which we can then obtain the coefficients  $\alpha_j$  by taking the scalar product with  $\mathbf{w}_i$

$$\alpha_i = \frac{\langle \mathbf{w}_i, \mathbf{f} \rangle}{\omega_i^2 - \omega^2} \quad (5.7)$$

and using  $\langle \mathbf{w}_i, \underline{\mathbf{M}}\mathbf{w}_j \rangle = \delta_{ij}$  since the eigenvectors stemming from a hermitian eigenvalue problem can be made M-orthonormal.

Inspecting (5.7), we see that if  $\langle \mathbf{w}_i, \mathbf{f} \rangle = 0$ , for one of the eigenpairs  $\{\mathbf{w}_i, \omega_i^2\}$  then the resonant mode at  $\omega = \omega_i$  is suppressed, since  $\alpha_i = 0$ . Because neither the generalized eigenvalue problem nor the analytical considerations are “informed” about the nature of the source term  $\mathbf{f}$ , this “effect” is usually not accounted for. Even specifying as the initial iteration vector for the eigsh eigenvalue solver the finite element representation of the source term  $\mathbf{f}$  itself to avoid finding eigensolutions that are orthogonal to it did not prove to be effective.

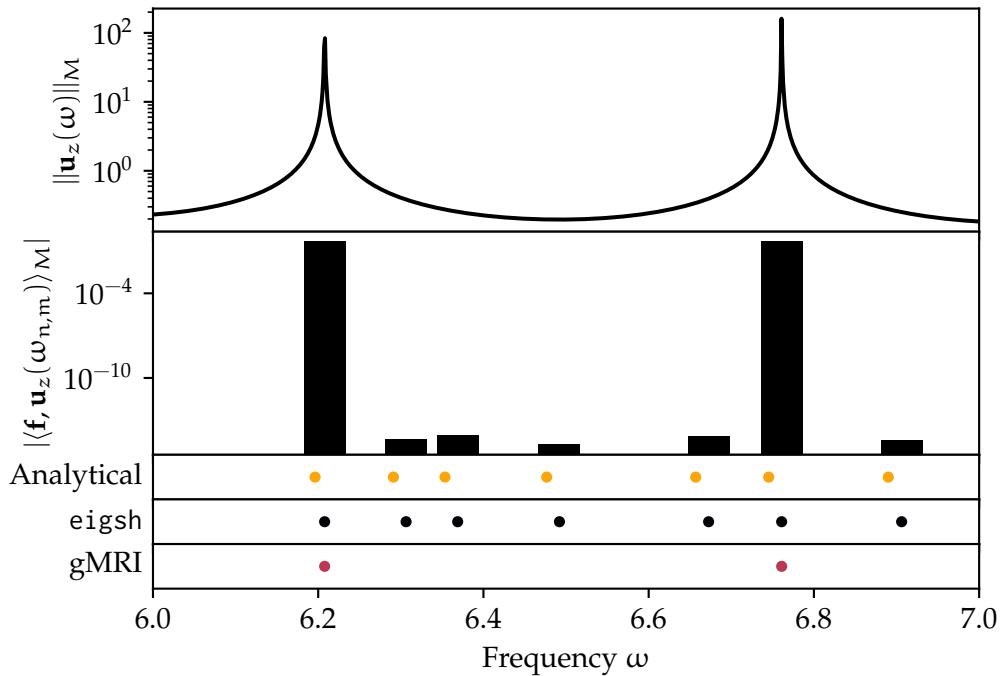


FIGURE 5.6 – For still the same problem but this time in a slightly higher frequency range  $\omega = [6, 7]$ , resonant frequencies determined using the three discussed approaches are shown (bottom). Only two resonant frequencies actually visibly emerge from the  $\|\cdot\|_M$ -norm spectrum computed with the FEM (top). Multiple resonant frequencies  $\omega_{n,m}$  are suppressed, which is the case whenever the corresponding solution  $\mathbf{u}_z(\omega_{n,m})$  happens to be orthogonal to the source vector  $\mathbf{f}$  (middle). gMRI successfully avoids yielding suppressed resonant frequencies while both the analytical and `eigsh` approach wrongly yield resonant frequencies that in actuality are suppressed.

### 5.1.3 APPROXIMATING SOLUTIONS ALONG A TRACE

I now restrict my view to approximating solutions  $\mathbf{u}_z|_{\Gamma}$  along a portion  $\Gamma$  of the boundary. Setting

$$\|\bar{\mathbf{u}}_z\|_{\underline{\mathbf{M}}(\Gamma)}^2 = \bar{\mathbf{u}}_z|_{\Gamma}^H \underline{\mathbf{M}}(\Gamma) \bar{\mathbf{u}}_z|_{\Gamma} \approx \int_{\Gamma} \|\mathbf{u}_z\|^2 \quad (5.8)$$

with  $\bar{\mathbf{u}}_z \in \mathbb{C}^N$  as the vertex basis representation of the two-dimensional solution  $\mathbf{u}_z$ , while  $\underline{\mathbf{M}}(\Gamma)$  is the matrix representation of the discretized integral form along  $\Gamma$ . Again, I will not bother putting bars everywhere and will instead just use  $\mathbf{u}_z$ . Running gMRI (Algorithm 2) on the restriction  $\mathbf{u}_z|_{\Gamma}$  instead of the full solution  $\mathbf{u}_z$  is a straight forward generalization of the algorithm. Nevertheless, when taking as  $\Gamma = \Gamma_N$  the inlet of the rectangular resonant cavity, no convergence in the relative error norm restricted to this trace appears to happen with gMRI (see Figure 5.7), and the spectral gap (4.21) becomes gigantic. This is due to the fact that the traces of the solutions at the boundary  $\Gamma_N$  are heavily linearly dependent.

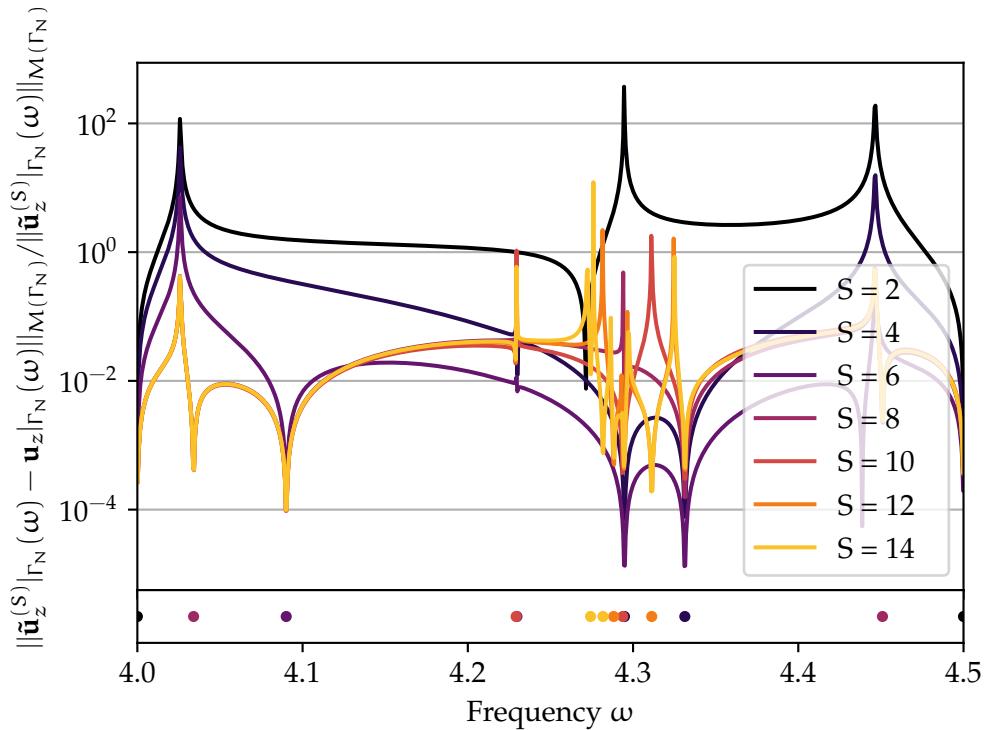


FIGURE 5.7 – The non-convergent behavior of the approximate relative error norm during every second iteration of gMRI of the solution restricted to the inlet  $\Gamma_N$ . Here,  $\tilde{\mathbf{u}}_z^{(S)}|_{\Gamma_N}$  stands for the rational surrogate for the solution on the trace built using  $S$  support points.

The symmetry may be broken by introducing a small “cubby” along the side of the cavity (see Figure 5.8).

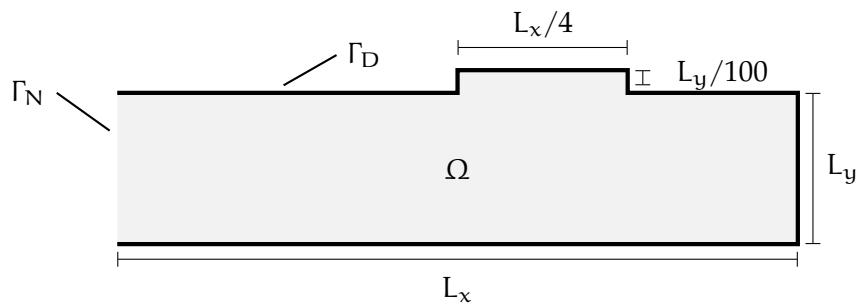


FIGURE 5.8 – A tiny cubby is introduced alongside one of the edges in order to break the linear dependence of solutions restricted to a trace. It extends along a quarter of the length of the cavity but extends outwards only by 1/100-th (sketch not drawn to scale!) of the height of the cavity.

Despite only adding a minor perturbation to the geometry of the cavity, the improved convergence behavior may immediately be seen in Figure 5.9.

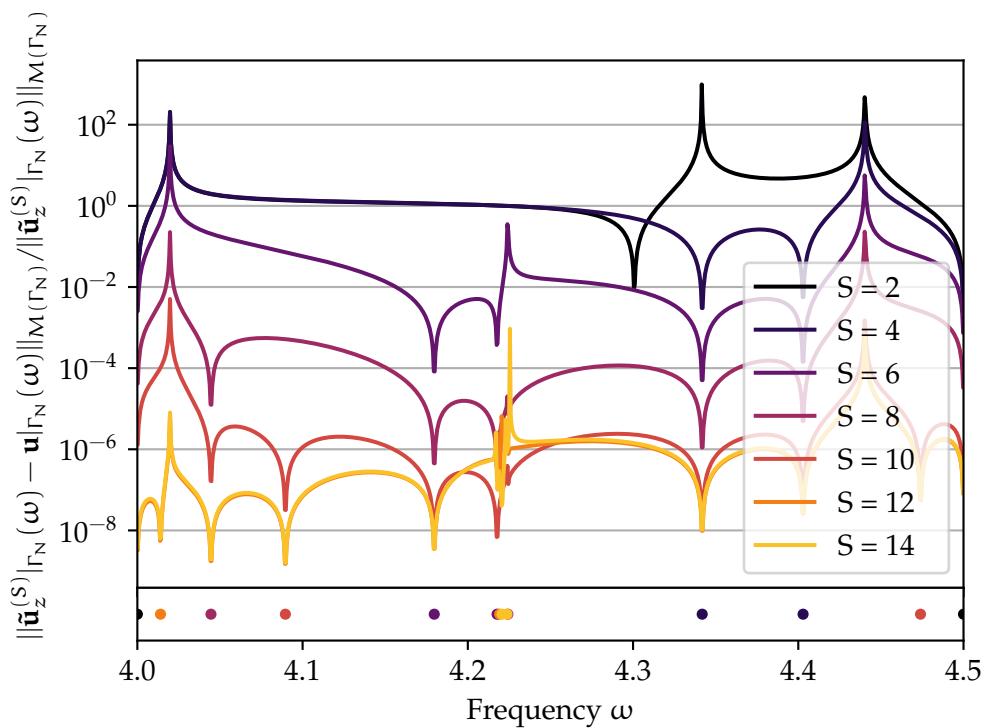


FIGURE 5.9 – Breaking the symmetry (i.e. the linear dependence of the trace  $\Gamma_N$  of the solution) with a tiny cubby (see Figure 5.8) proves to be quite an effective remedy for the stability of building the rational surrogate with gMRI. For every second iteration of the algorithm, an estimate for the relative error of the surrogate  $\tilde{u}_z^{(S)}|_{\Gamma_N}$  built with  $S$  support points (shown at the bottom) is plotted. Compared to Figure 5.7, the error goes down much more uniformly.

## 5.2 IMPERFECTLY CONDUCTING BOUNDARIES

A slight modification of the exact same problem as before in Section 5.1 (the derivation of this modification is derived in Section 2.3.2) brings us to the example of an imperfectly conducting boundary. At the edge opposite of the inlet (i.e.  $x = L_x$ ), the impedance is first set to  $\lambda = 1$ .

A solution to the system is plotted in Figure 5.10 and the  $\|\cdot\|_M$ -norm for both the FEM and gMRI are plotted in Figure 5.11.

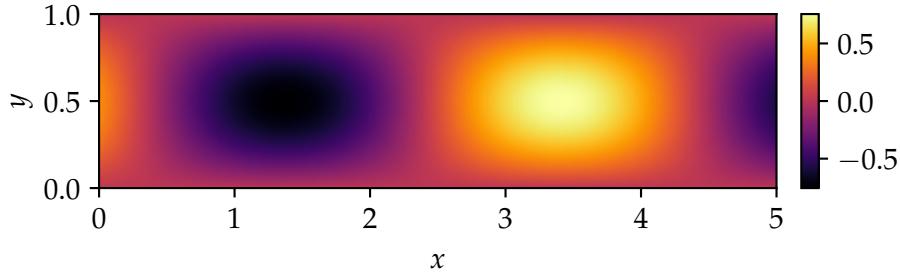


FIGURE 5.10 – The solution  $\mathbf{u}_z$  obtained with the FEM at the frequency  $\omega = 3.5$ . Unlike in Figures 5.2 and 5.3, the solution does no longer have to vanish at the imperfect boundary on the right-hand edge.

To find resonant frequencies for this system numerically, the nonlinear eigenproblem

$$(\underline{\mathbf{K}} - i\omega \underline{\mathbf{I}} - \omega^2 \underline{\mathbf{M}}) \mathbf{w} = \mathbf{0} \quad (5.9)$$

with  $\mathbf{I}$  being the matrix representation corresponding to the form

$$\int_{\Gamma_I} \lambda u_z v_z \quad (5.10)$$

can be solved.

Unfortunately, solving this eigenvalue problem is not as straight forward as the one that was encountered in the previous example. Likely the simplest approach to solving (5.9) is to linearize the system by defining  $\mathbf{v} = \omega \mathbf{w}$ , such that we can convert it to a system twice the size that reads

$$\begin{bmatrix} \underline{\mathbf{K}} & \underline{\mathbf{1}} \\ -i\underline{\mathbf{I}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} = \omega \begin{bmatrix} \underline{\mathbf{1}} & \underline{\mathbf{M}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} \quad (5.11)$$

Usually, however, the two matrices involved in the eigenproblem are no longer going to be hermitian. Combined with the fact that the problem is now twice the size of the original one, solving the system with the sparse eigenvalue solver `eigs` included in the `scipy` library will cost more computational resources than gMRI does (see Table 5.2 for the comparison of the time spent in computation between the two methods).

The effect of the impedance  $\lambda$  on the position of the (complex) resonant frequencies is visualized in Figure 5.12. For  $\lambda \rightarrow 0$  we recover a Neumann boundary condition, while for  $\lambda \rightarrow \infty$ , a Dirichlet-type boundary is mimicked.

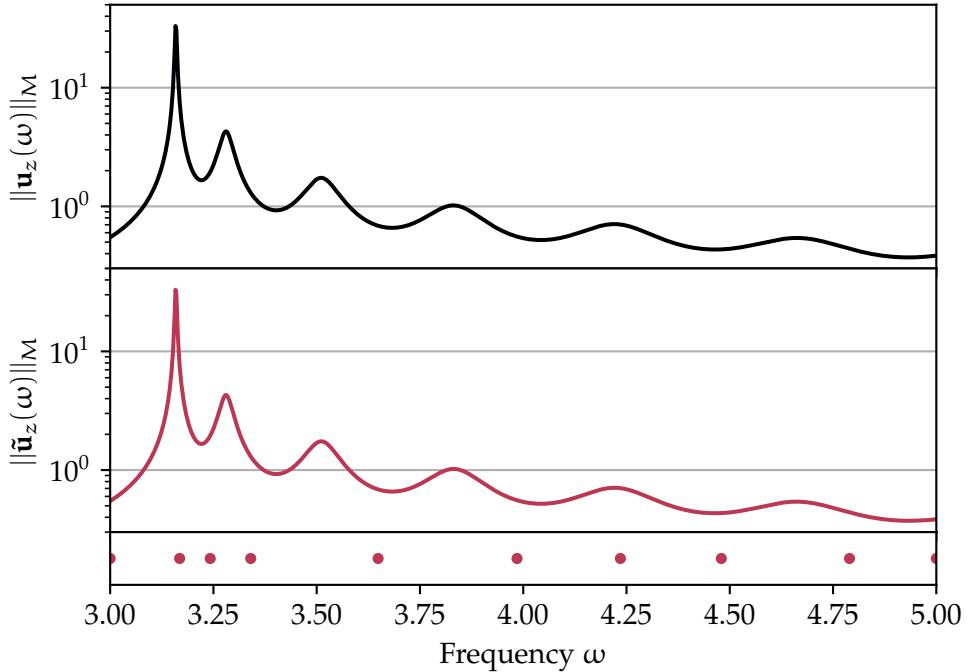


FIGURE 5.11 – The  $\|\cdot\|_{\mathcal{M}}$ -norm of the FEM solution  $\mathbf{u}_z(\omega)$  (top) and its surrogate  $\tilde{\mathbf{u}}_z(\omega)$  (middle) for the rectangular cavity with an imperfectly conducting edge is shown. The locations of the greedy support points are indicated (bottom).

TABLE 5.2 – Comparison of the computation times for identifying the resonant frequencies with a real part in the interval  $\omega = [3, 5]$  between eigs and gMRI with a tolerance of  $\tau = 10^{-2}$  and 1000 linearly spaced candidate support points in  $\omega = [3, 5]$ . To get a full comparison to the equivalent computations in Table 5.1 an attempt was made to use eigs for solving the problem with 745513 DOFs, but proved to be unsuccessful due to a memory overflow.

	eigs	gMRI
DOF	t	t
713	$57.8 \pm 2.35$ ms	$62.8 \pm 0.8$ ms
7412	$861.0 \pm 42.4$ ms	$498.0 \pm 11.7$ ms
74722	$21.8 \pm 1.1$ s	$5.9 \pm 0.3$ s

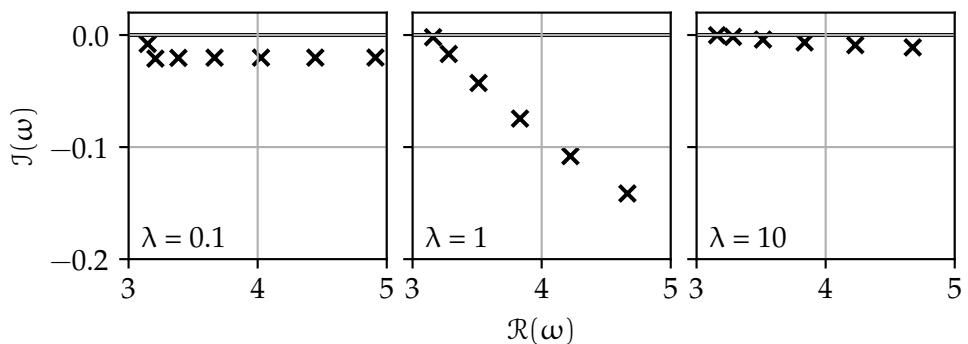


FIGURE 5.12 – Resonant frequencies of a rectangular cavity with an imperfect boundary with impedance  $\lambda \in \{0.1, 1, 10\}$ . Spurious resonant frequency far away from the relevant interval are not of interest for this problem and hence cropped off.

### 5.3 DUAL MODE CIRCULAR WAVEGUIDE FILTER

The Dual-Mode Circular Waveguide Filter (DMCWF) depicted in Figure 5.13 is studied. It is a highly symmetric three-dimensional waveguide that consists of two opposing ports connected by a cylinder that is split in two by a cross iris. Breaking this symmetry are merely four screws: The two horizontal tuning screws as well as the two coupling screws. A brief theoretical treatment is given in Section 2.3.3.

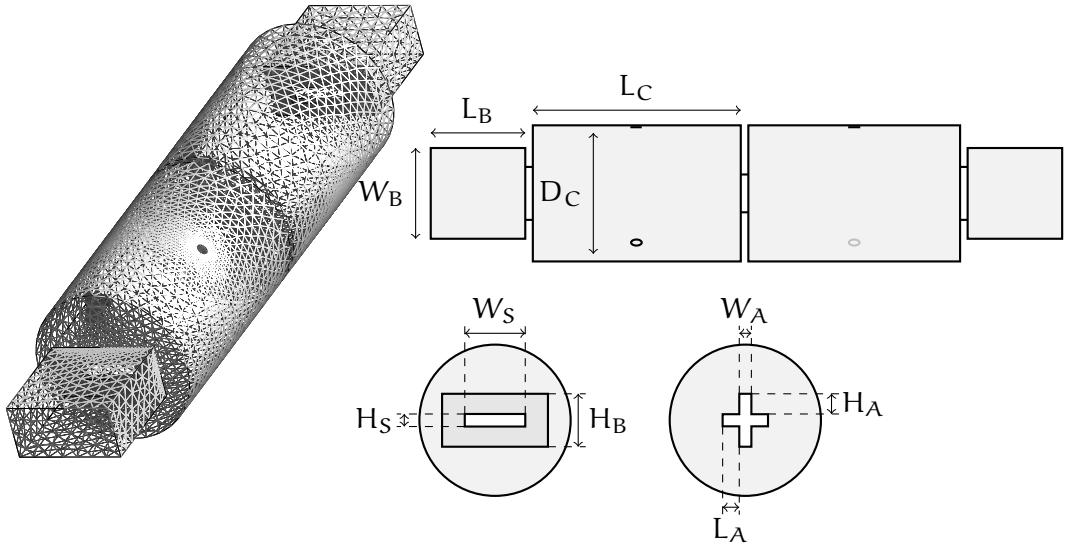


FIGURE 5.13 – The surface-mesh of the modelled dual-mode circular waveguide filter (left). An effort was made to keep the dimensions as similar to the ones given in [9], but a full correspondence cannot be guaranteed due to the author being unable to resolve certain ambiguities that were encountered. The model adheres to the following dimensions:  $L_C = 43.87$  mm,  $D_C = 28.0$  mm,  $L_B = 20.0$  mm,  $W_B = 19.05$  mm,  $H_B = 9.525$  mm,  $W_S = 10.05$  mm,  $H_S = 3.0$  mm,  $W_A = 2.0$  mm,  $H_A = 3.375$  mm,  $L_A = 2.825$  mm; The thickness of all irises is 2.0 mm; The screws are placed exactly halfway up each of the two resonant cylinders, with the horizontal tuning screws reaching a depth of 3.82 mm into the cavity and coupling screws at angles  $\pm 45^\circ$  with a depth of 3.57 mm. Observe that the DMCWF is point symmetric with respect to the center of the cross iris.

I created the model in the computer-aided design modeler software application FreeCAD<sup>2</sup>. The mesh was defined using the 3D finite element mesh generator Gmsh<sup>3</sup> and is available for download on the git repository of this project [7]. In Gmsh, a element size factor of 0.2 and 5 smoothing steps are used for the Delaunay 3D meshing algorithm. Additionally, the mesh was refined around critical components such as the screws and irises using transfinite curves (the sharp-eyed reader may observe these refinements in the surface mesh visualized in Figure 5.13). A total of 185726 DOFs are accounted for. The conversion of the mesh to the FEniCS supported

<sup>2</sup><https://www.freecadweb.org/>

<sup>3</sup><https://gmsh.info/>

.xml format was performed with the open source mesh format converter meshio<sup>4</sup>.

In the following, I would like to demonstrate the capability of the gMRI algorithm to approximate scattering coefficients for the DMCWF. The definition of the scattering matrix is a adaptation from [13] to

$$\underline{\mathbf{S}}(\omega) = \underline{\mathbf{1}} - 2 \left( \underline{\mathbf{1}} + i \frac{\omega}{2\pi} \sqrt{\frac{\mu}{\epsilon}} \sqrt{\frac{1 - (\omega_c/\omega_0)^2}{1 - (\omega_c/\omega)^2}} \underline{\mathbf{F}}^H \underline{\mathbf{U}}(\omega) \right)^{-1} \quad (5.12)$$

where  $\underline{\mathbf{F}} = [\mathbf{f}_1, \mathbf{f}_2]$  and  $\underline{\mathbf{U}} = [\mathbf{u}_1, \mathbf{u}_2]$  with  $\mathbf{f}_1$  the source term resulting when the DMCWF is forced from one side, producing the solution  $\mathbf{u}_1$ , and the  $\mathbf{f}_2$  when forcing from the other side to produce  $\mathbf{u}_2$ .  $\omega_c = 4.122 \times 10^{10}$  and  $\omega_0 = 6.283 \times 10^{10}$  were assumed. The scattering coefficients  $S_{ij}$  are then precisely the entries of the scattering matrix

$$\underline{\mathbf{S}}(\omega) = \begin{bmatrix} S_{11}(\omega) & S_{12}(\omega) \\ S_{21}(\omega) & S_{22}(\omega) \end{bmatrix} \quad (5.13)$$

For the simulation,  $\epsilon = 4\pi \times 10^{-7}$  and  $\mu = 8.854187 \times 10^{-12}$  were assumed. The one-sided forcing was performed with  $\mathbf{g}|_{\Gamma_L} = -\mathbf{e}_z$  from the left-hand inlet  $\Gamma_L$ , and  $\mathbf{g}|_{\Gamma_R} = \mathbf{e}_z$  from the right-hand inlet  $\Gamma_R$ . The scattering coefficients  $S_{11}(\omega)$  and  $S_{21}(\omega)$  once computed by using the solutions at 150 uniformly spaced sample frequencies directly obtained with the FEM, and once from the rational surrogate computed with gMRI at a tolerance of  $\tau = 10^{-2}$  and 1000 uniformly spaced candidate test points in the frequency range  $\omega \in [7.257 \times 10^{10}, 7.508 \times 10^{10}]$ . Both are shown (but only one is seen) in Figure 5.14. At somewhat more than four minutes, the gMRI was able to deliver the approximated scattering coefficients roughly 25 times faster than the FEM. Compared to [3], the passband is similar in shape but slightly shifted towards the left, i.e. lower frequencies.

The progression of the absolute error with the number of support points used to build the surrogate can be seen in Figure 5.15.

---

<sup>4</sup><https://github.com/nschloe/meshio>

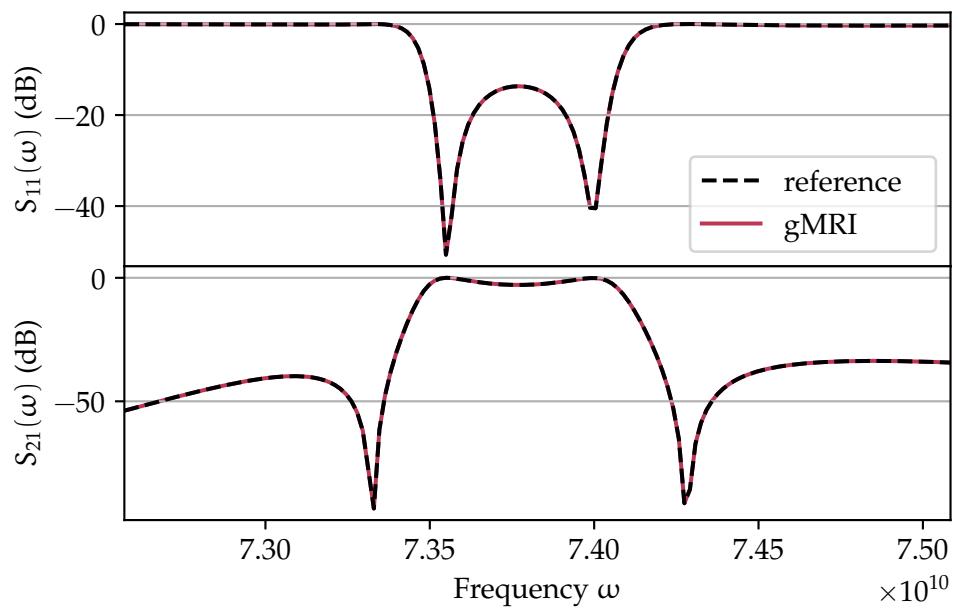


FIGURE 5.14 – The scattering coefficients  $S_{11}(\omega)$  and  $S_{21}(\omega)$  defined in (5.13) computed as a reference from solving the system at discrete sample points and the FEM, as well as the ones obtained by using gMRI.

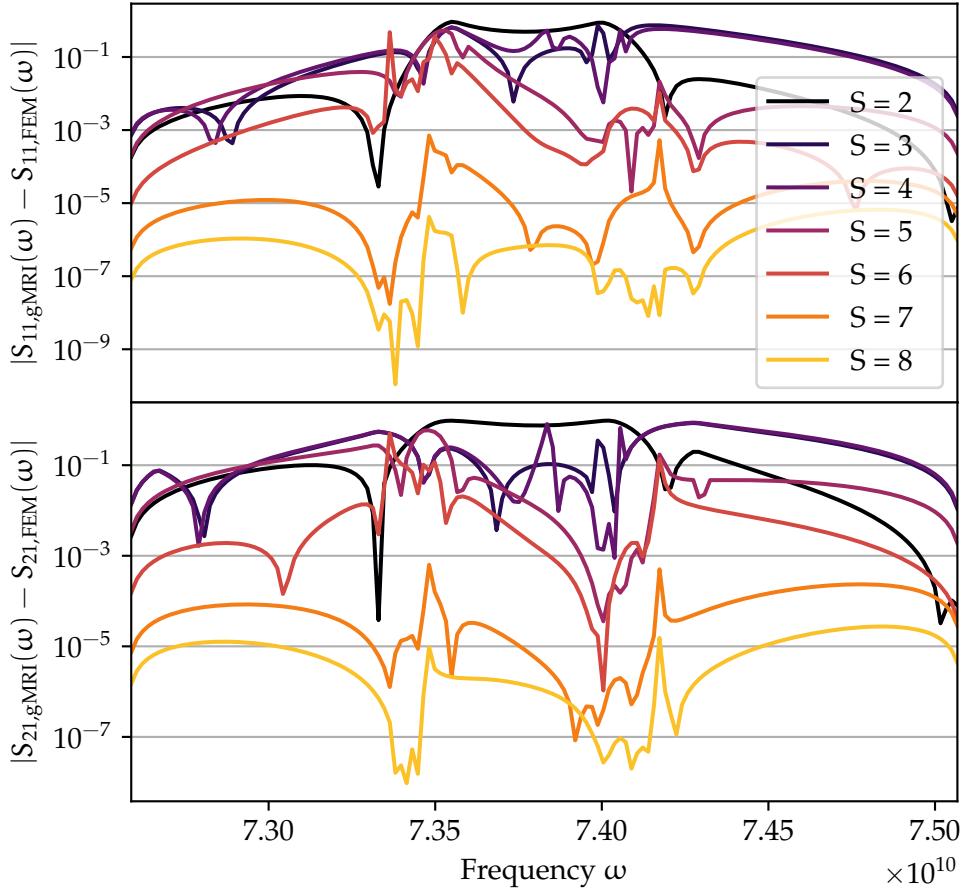


FIGURE 5.15 – Progression of the absolute error between the reference scattering coefficients ( $S_{11,FEM}$  and  $S_{21,FEM}$ ) obtained by solving the problem with FEM and the ones obtained by using gMRI ( $S_{11,gMRI}^{(S)}$  and  $S_{21,gMRI}^{(S)}$ ) with  $S$  support points (please excuse the unfortunate notation). In reality, computing the rational surrogate when forcing from the right-hand side required one more support point to meet the stopping criterion in gMRI than the surrogate for forcing from the left-hand side. However, because both right- and left-hand side are required to compute  $S_{11,gMRI}^{(S)}$  and  $S_{21,gMRI}^{(S)}$ , this figure ignores the very last iteration  $S = 9$ , where only a right-hand surrogate has been computed.

## 6 CONCLUSION AND OUTLOOK

In the above report I have studied the suitability of the Minimal Rational Interpolation (MRI) method and particularly the Greedy Minimal Rational Interpolation (gMRI) algorithm for the time-harmonic Maxwell's equations. In general, the gMRI algorithm paired with the Finite Element Method (FEM) offer an efficient, reliable and flexible way of producing an approximation of the frequency dependence of the vector potential  $\mathbf{u}$  in problems of the form (2.10).

Finding resonant frequencies of a simple system with the help of gMRI was found to be comparable to conventional approaches when considering the accuracy and time spent in computation (cf. Section 5.1 and particularly Table 5.1). In terms of ease of use, gMRI proved to exhibit the key property of not incorrectly identifying resonant frequencies that are suppressed in the case of a highly symmetric model (Figure 5.6). Furthermore, with gMRI no a priori knowledge of the number of resonant frequencies is required, which is not the case with the conventional approach the method was compared to.

Furthermore, specifically for larger and more complex systems, e.g. the nonlinear eigenproblem solved in Section 5.2 or the treatment of the Dual-Mode Circular Waveguide Filter (DMCWF) in Section 5.3, excelled in its performance mainly due to its scalability and time efficiency.

Stability issues of MRI mainly emerging with highly symmetric meshes and when using as the snapshots only the solution along a trace were identified in Section 5.1.3 and remedies were suggested and tested to be effective in resolving this issue.

Further work would have to be put into the study of the DMCWF, after a proper verification of both the dimensions and the meshing with a reference model has been conducted. It would also be interesting to study gMRI for a wider class of time-harmonic Maxwell problems, for instance with a non-trivial current density  $\mathbf{j}$  or systems with spatially dependent  $\epsilon$  and  $\mu$ . Using the implementations I have published in [7], these extensions are made completely straightforward to add into the code.

## REFERENCES

- [1] F. Bonizzoni, F. Nobile, and I. Perugia. Convergence analysis of padé approximations for helmholtz frequency response problems. *ESAIM: M2AN*, 52(4): 1261 – 1284, 2018. doi: 10.1051/m2an/2017050.
- [2] F. Bonizzoni, D. Pradovera, and M. Ruggeri. Rational-based model order reduction of helmholtz frequency response problems with adaptive finite element snapshots. 2021. doi: 10.48550/arXiv.2112.04302.
- [3] V. de la Rubia, U. Razafison, and Y. Maday. Reliable fast frequency sweep for microwave devices via the reduced-basis method. *IEEE Transactions on Microwave Theory and Techniques*, 57(12):2923 – 2937, 2009. doi: 10.1109/TMTT.2009.2034208.
- [4] F. Kagerer. Finite elements for maxwell's equations. Bsc thesis, Johannes Kepler University Linz, 2018.
- [5] G. Klein. Applications of linear barycentric rational interpolation. <https://core.ac.uk/download/pdf/20659062.pdf>, 2012.
- [6] H. P. Langtangen and A. Logg. *Solving PDEs in Python: The FEniCS Tutorial I*. Springer, 2016. ISBN 978-3-319-52461-0. doi: 10.1007/978-3-319-52462-7.
- [7] F. Matti. Minimal rational interpolation for time-harmonic maxwell's equations. <https://github.com/FMatti/Maxwell-MRI>.
- [8] P. Monk. *Finite Element Methods for Maxwell's Equations*. Oxford Science Publications, 2003. ISBN 0-19-850888-3.
- [9] J. R. Montejo-Garai and J. Zapata. Full-wave design and realization of multi-coupled dual-mode circular waveguide filters. *IEEE Transactions on Microwave Theory and Techniques*, 43(6):1290 – 1291, 1995.
- [10] F. Nobile. Lecture notes of the course numerical approximation of partial differential equations - i, 2017-2018.
- [11] D. Pradovera. Interpolatory rational model order reduction of parametric problems lacking uniform inf-sup stability. *SIAM Journal on Numerical Analysis*, 58 (4):2265–2293, 2020. doi: 10.1137/19M1269695.
- [12] D. Pradovera. *Model order reduction based on functional rational approximants for parametric PDEs with meromorphic structure*. PhD thesis, EPFL, 2021. Thèse n° 9271.
- [13] D. Pradovera and F. Nobile. Frequency-domain non-intrusive greedy model order reduction based on minimal rational approximation. pages 159–167, 2021. doi: 10.1007/978-3-030-84238-3\_16.
- [14] L. N. Trefethen. Householder triangularization of a quasimatrix. *IMA Journal of Numerical Analysis*, 30(4):887–897, 2010. doi: 10.1093/imanum/drp018.

## APPENDIX

This section contains some derivations that did not end up making it into the report, but were too much fun not to include them anyway. Most of them make use of the Levi-Civita tensor. This completely antisymmetric tensor  $\varepsilon_{ijk}$  is fully characterized by the following defining properties:

$$\begin{cases} \varepsilon_{123} = 1 & \text{gauge} \\ \varepsilon_{ijk} = 0 \text{ if } |\{i,j,k\}| < 3 & \text{cardinality} \\ \varepsilon_{ijk} = \varepsilon_{jki} & \text{cyclic permutation} \\ \varepsilon_{ijk} = -\varepsilon_{ikj} & \text{non-cyclic permutation} \end{cases} \quad (6.1)$$

It may be employed to rewrite the components of the curl of a vector-function  $\mathbf{a}$  as the sum

$$(\nabla \times \mathbf{a})_k = \sum_i \sum_j \varepsilon_{ijk} \partial_i a_j \quad (6.2)$$

where  $\partial_i$  denotes the partial derivative with respect to the  $i$ -th coordinate direction.

### DETAILED DERIVATION FOR THE WEAK FORMULATION OF THE TIME-HARMONIC POTENTIAL EQUATION

The goal is to rewrite the curl-integral on the left-hand side of (2.12):

$$\int_{\Omega} \langle \nabla \times (\mu^{-1} \nabla \times \mathbf{u}), \mathbf{v} \rangle \quad (6.3)$$

In order to simplify the curls and apply the Gauss theorem, I first show the following vector calculus identity:

Curl product rule

$$\langle \nabla \times \mathbf{a}, \mathbf{b} \rangle = \langle \nabla, \mathbf{a} \times \mathbf{b} \rangle + \langle \mathbf{a}, \nabla \times \mathbf{b} \rangle \quad (6.4)$$

where  $\mathbf{a}, \mathbf{b}$  are vector-valued functions. Rewriting the left-hand side of this equation by using the Levi-Civita tensor (6.2) yields

$$\begin{aligned} \langle \nabla \times \mathbf{a}, \mathbf{b} \rangle &= \sum_k (\nabla \times \mathbf{a})_k b_k \\ &= \sum_k \left( \sum_i \sum_j \varepsilon_{ijk} \partial_i a_j \right) b_k \\ &= \sum_k \sum_i \sum_j \partial_i (\varepsilon_{ijk} a_j b_k) - \sum_k \sum_i \sum_j a_j (\varepsilon_{ijk} \partial_i b_k) \\ &= \sum_k \sum_i \sum_j \partial_i (\varepsilon_{jki} a_j b_k) - \sum_k \sum_i \sum_j a_j ((-\varepsilon_{ikj}) \partial_i b_k) \\ &= \sum_i \partial_i (\mathbf{a} \times \mathbf{b})_i + \sum_j a_j (\nabla \times \mathbf{b})_j \\ &= \langle \nabla, \mathbf{a} \times \mathbf{b} \rangle + \langle \mathbf{a}, \nabla \times \mathbf{b} \rangle \end{aligned} \quad (6.5)$$

by expressing the scalar product as a component-sum, using the product rule and applying the symmetry and anti-symmetry properties of the Levi-Civita tensor. Now the identity (6.4) to (6.3) together with Gauss' theorem gives

$$\begin{aligned}\int_{\Omega} \langle \nabla \times (\mu^{-1} \nabla \times \mathbf{u}), \mathbf{v} \rangle &= \int_{\Omega} \langle \nabla, (\mu^{-1} \nabla \times \mathbf{u}) \times \mathbf{v} \rangle + \int_{\Omega} \langle \mu^{-1} \nabla \times \mathbf{u}, \nabla \times \mathbf{v} \rangle \\ &= \int_{\partial\Omega} \langle (\mu^{-1} \nabla \times \mathbf{u}) \times \mathbf{v}, \mathbf{n} \rangle + \int_{\Omega} \langle \mu^{-1} \nabla \times \mathbf{u}, \nabla \times \mathbf{v} \rangle\end{aligned}\quad (6.6)$$

For later convenience, the boundary integral can further be simplified using the

Commutative behavior of the scalar triple product

$$\langle \mathbf{a} \times \mathbf{b}, \mathbf{c} \rangle = -\langle \mathbf{a} \times \mathbf{c}, \mathbf{b} \rangle \quad (6.7)$$

This identity follows immediately from a small manipulation with the Levi-Civita tensor:

$$\begin{aligned}\langle \mathbf{a} \times \mathbf{b}, \mathbf{c} \rangle &= \sum_k \left( \sum_i \sum_j \epsilon_{ijk} a_i b_j \right) c_k \\ &= \sum_j \left( \sum_i \sum_k (-\epsilon_{ikj}) a_i c_k \right) b_j \\ &= -\langle \mathbf{a} \times \mathbf{c}, \mathbf{b} \rangle\end{aligned}\quad (6.8)$$

The boundary integral becomes

$$\int_{\partial\Omega} \langle (\mu^{-1} \nabla \times \mathbf{u}) \times \mathbf{v}, \mathbf{n} \rangle = - \int_{\partial\Omega} \langle (\mu^{-1} \nabla \times \mathbf{u}) \times \mathbf{n}, \mathbf{v} \rangle \quad (6.9)$$

This concludes the short derivation, because now (6.3) may be rewritten as

$$-\int_{\partial\Omega} \langle (\mu^{-1} \nabla \times \mathbf{u}) \times \mathbf{v}, \mathbf{n} \rangle + \int_{\Omega} \langle \mu^{-1} \nabla \times \mathbf{u}, \nabla \times \mathbf{v} \rangle \quad (6.10)$$

## INVARIANCE OF REPEATED CROSS PRODUCTS

I would like to demonstrate the

Invariance of repeated cross products

If  $\mathbf{n} \perp \mathbf{u}$  and  $\|\mathbf{n}\| = 1$ , then

$$(\mathbf{n} \times \mathbf{u}) \times \mathbf{n} = \mathbf{u} \quad (6.11)$$

I again resort to the old faithful Levi-Civita tensor which satisfies the identity

$$\sum_i \epsilon_{jki} \epsilon_{lmi} = \delta_{jl} \delta_{km} - \delta_{jm} \delta_{kl} \quad (6.12)$$

Furthermore, I make use of the invariance of the tensor under cyclic permutation of the indices (according to Equation (6.1)) to obtain

$$\begin{aligned}
[(\mathbf{n} \times \mathbf{u}) \times \mathbf{n}]_k &= \sum_i \sum_j \varepsilon_{ijk} (\mathbf{n} \times \mathbf{u})_i n_j \\
&= \sum_i \sum_j \varepsilon_{ijk} \sum_l \sum_m \varepsilon_{lmi} n_l u_m n_j \\
&= \sum_i \sum_j \sum_l \sum_m \varepsilon_{jki} \varepsilon_{lmi} n_l u_m n_j \\
&= \sum_j \sum_l \sum_m (\delta_{jl} \delta_{km} - \delta_{jm} \delta_{kl}) n_l u_m n_j \\
&= \sum_j n_j u_k n_j - \sum_j n_k u_j n_j \\
&= \|\mathbf{n}\|^2 u_k - \langle \mathbf{u}, \mathbf{n} \rangle n_k \\
&= u_k
\end{aligned} \tag{6.13}$$

which concludes the component-wise proof.