EPFL

SEMESTER PROJECT

Minimal Rational Interpolation for Time-Harmonic Maxwell's Equations

Fabio Matti

supervised by
Prof. Fabio Nobile
Dr. Davide Pradovera

May 25, 2022

ABSTRACT

Minimal Rational Interpolation (MRI) provides an efficient and reliable way to approximate the dependence of a characteristic quantity of a model on one of its parameters. The focus of this report is put on the Greedy Minimal Rational Interpolation (gMRI) algorithm and particularly on way to enhance its performance. This algorithm is then applied to three example problems concerning the time-harmonic Maxwell's equations in the frequency domain. A brief evaluation of the advantages and disadvantages of gMRI as compared to conventional approaches for finding quantities, such as resonant modes, of interest for problems of this type is conducted.

CONTENTS

1 INTRODUCTION

A wide class of problems in physics and engineering concerns itself with the study of the dependence of a model on one of its parameters. Of interest is usually a characteristic quantity that covaries with said 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 the continuous dependence of a model on one of its parameters. The approach has proven effective and efficient (both memory- and computation wise) in applications on Helmholtz-type problems [3, 1].

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 a single curl-curl equation. 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).

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 multiple values of the model parameter that is of interested. 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, ρ an electric charge density, and $\mathbf{j}$ an electric current density. Maxwell's equations are stated in [8] as

$$\nabla \cdot (\epsilon \mathbf{E}) = \rho \tag{2.1}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{2.2}$$

$$\nabla \times \mathbf{E} = -\partial_t \mathbf{B} \tag{2.3}$$

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

with ε being the permittivity and μ the permeability (whose names let alone their values I always tend to forget).

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 ϕ, referred to as the scalar potential.

The physical quantities $\mathbf{E}$ and $\mathbf{B}$ remain unchanged if we transform $\mathbf{u} \to \mathbf{u}' = \mathbf{u} + \nabla\psi$ or $\phi \to \phi' = \phi - \partial_t \psi$ for arbitrary functions ψ. A convenient choice of ψ is suggested in [6] to be

$$\psi = \int_0^t \phi \, dt' \tag{2.5}$$

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

$$\mathbf{E} = -\partial_t \mathbf{u} \tag{2.6}$$

$$\mathbf{B} = \nabla \times \mathbf{u} \tag{2.7}$$

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

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

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

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

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

Time-harmonic potential equation

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

2.2 Weak formulation for the time-harmonic potential equation

Equation (2.10) may be multiplied by a vector-valued function $\mathbf{v} \in H_{\text{curl}}(\Omega)$, where

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

and then integrated over the whole computational domain Ω to obtain

$$\int_\Omega (\nabla \times (\mu^{-1} \nabla \times \mathbf{u})) \cdot \mathbf{v} - \omega^2 \int_\Omega \epsilon \mathbf{u} \cdot \mathbf{v} = \int_\Omega \mathbf{j} \cdot \mathbf{v} \tag{2.12}$$

This may further be simplified (2.12) to (I allow myself to spare you the details of this computation, but put a proper derivation in an appendix at the end of the report):

Weak formulation of the time-harmonic potential equation

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

where $\mathbf{n}$ denotes the surface normal to the boundary $\partial\Omega$ of the computational domain Ω.

Boundary conditions on the electric field $\mathbf{E}$ may be most easily enforced in a Dirichlet-type fashion through the relation (2.6) and the assumption (2.9)

$$\mathbf{u}|_{\Gamma_D} = -\frac{1}{i\omega} \mathbf{E}|_{\Gamma_D} \tag{2.14}$$

Those on the magnetic field $\mathbf{B}$ through a Neumann-type condition following from (2.7) and again (2.9)

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

2.3 Examples

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

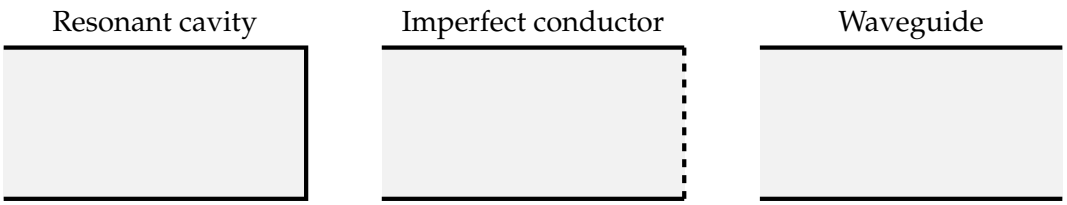


FIGURE 2.1 – Schematic visualization of the most trivial case for each of the boundary configurations that will be analyzed in Section 5. The perfectly conducting boundaries are drawn in black, while the imperfectly conducting boundary appears dashed. Inlets and exits are left unmarked.

2.3.1 TWO-DIMENSIONAL RESONANT CAVITY

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

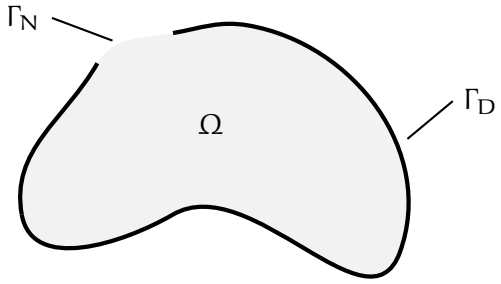


FIGURE 2.2 – An abstract example of a two-dimensional resonant cavity enclosing a domain Ω with a perfectly conducting boundary Γ_D and featuring a single inlet Γ_N.

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 (**??**) simplifies to the scalar product of two gradients:

$$(\mu^{-1}\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v}) = (\mu^{-1}\nabla u_z) \cdot (\nabla v_z) \tag{2.16}$$

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

$$\int_\Omega (\mu^{-1}\nabla u_z) \cdot (\nabla v_z) - \omega^2 \int_\Omega \epsilon u_z v_z = \int_{\partial\Omega} g_z v_z \tag{2.17}$$

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 Γ_D, treated in [8], it holds that

$$\mathbf{n} \times \mathbf{E} = 0, \text{ on } \Gamma_D \tag{2.18}$$

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 $\mathbf{u}|_{\Gamma_D} = 0$ 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):

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

2.3.2 IMPERFECT CONDUCTOR

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

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

with a parameter $\lambda > 0$ 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 (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)

$$g_z = i\omega\lambda u_z \text{ on } \Gamma_D \tag{2.21}$$

Therefore, an impedance boundary can be treated in almost the same way as a Neumann boundary in the two-dimensional weak formulation (2.17) of a resonant cavity.

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 Γ_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 "exit" Γ_e, we set $\mathbf{g}|_{\Gamma_e} = \mathbf{0}$.

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 shape

$$\text{Find } \mathbf{u} \in H_{\text{curl}}(\Omega), \text{ such that } a_\omega(\mathbf{u}, \mathbf{v}) = L(\mathbf{v}), \ \forall \mathbf{v} \in H_{\text{curl}}(\Omega) \tag{3.1}$$

with the bilinear form

$$a_\omega(\mathbf{u}, \mathbf{v}) = \int_\Omega (\mu^{-1} \nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v}) - \omega^2 \int_\Omega \epsilon \mathbf{u} \cdot \mathbf{v} \tag{3.2}$$

and the linear form

$$L(\mathbf{u}) = \int_\Omega \mathbf{j} \cdot \mathbf{v} + \int_{\partial\Omega} \mathbf{g} \cdot \mathbf{v} \tag{3.3}$$

and the appropriate Hilbert space $H_{\text{curl}}(\Omega)$ defined in (2.11).

A sequence of appropriate finite dimensional spaces $H_{\text{curl},h}(\Omega)$ is introduced, and the Galerkin problem is then formulated as (see [9] for details)

Galerkin problem for the time-harmonic potential equation

$$\text{Find } \mathbf{u}_h \in H_{\text{curl},h}(\Omega), \text{ such that } a_\omega(\mathbf{u}_h, \mathbf{v}_h) = L(\mathbf{v}_h), \ \forall \mathbf{v} \in H_{\text{curl},h}(\Omega) \tag{3.4}$$

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

3.2 NUMERICAL APPROXIMATION OF PDES USING FENICS

FEniCS[1] bundles a collection of Python modules designed to automate solving a Partial Differential Equation (PDE). Inspired by the demonstrations encountered in [4], I will now guide you through a simple example, relevant to the context of this report, in order to show how the process of obtaining approximate solutions to PDEs with FEniCS.

Consider the time-harmonic potential equation (2.10) with the computational domain Ω being a cubic cavity with an inlet Γ_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` and `matplotlib.pyplot` for array manipulation and visualization respectively.

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

[1] https://fenicsproject.org/

A mesh for the cubic cavity Ω is generated by dividing the cube into a $10\times10\times10$ grid, whose cells are again subdivided into tetrahedrons.

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

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

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

The inlet is introduced at $x = 0$.

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

All other boundaries are perfectly conducting walls.

```
class PECWalls(fen.SubDomain):
    def inside(self, x, on_boundary):
        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.

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

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

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

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

```
g_N = fen.Expression(('0.0', '0.0', '1.0'), degree=2)
ds = fen.Measure('ds', subdomain_data=boundary_id)
```

Trial and test functions for the function space $H_{curl,h}(\Omega)$ are instantiated.

```
u = fen.TrialFunction(V)
v = fen.TestFunction(V)
```

The linear form (3.3) is assembled.

```
N = fen.assemble(fen.dot(g_N, 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.

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

The mass matrix (i.e. the second term in the bilinear form (3.2)) is assembled, and the Dirichlet boundary conditions are accounted for by setting all rows and columns corresponding to degrees of freedom on the perfectly conducting boundary to zero.

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

A function to compute an approximation of the $L_2(\Omega)$-norm of a solution to the system can be created.

```
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).

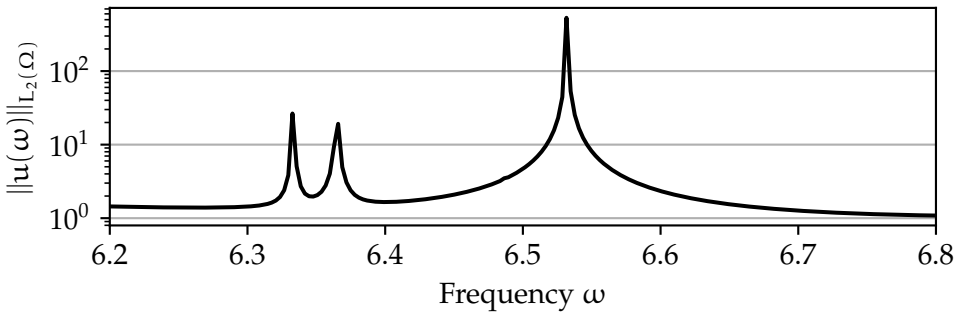


FIGURE 3.1 – Frequency response in the $L_2(\Omega)$-norm of a cubic cavity with one face acting as an inlet and all others as perfectly conducting boundaries. At resonant frequencies, the $L_2(\Omega)$-norm theoretically tends to infinity. Numerically, they appear as finite peaks in the frequency response.

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

Let $\mathbf{u} : \mathbb{C} \to \mathbb{C}^3$. Given "snapshots" of the function $\mathbf{u}(\omega_j)$ at ω_j for $j \in \{1, \dots, S\}$, the goal is to find a surrogate that locally (i.e. near $\omega_1, \dots, \omega_S$) satisfies

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

This may be achieved using the Minimal Rational Interpolation (MRI) technique, 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

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

Writing the double-curl operator in terms of a matrix $\underline{\mathbf{A}}$ allows for an expression of the solution $\mathbf{u}$ to (4.2) as

$$\mathbf{u} = (\underline{\mathbf{A}} - \omega^2)^{-1} \mathbf{j} \tag{4.3}$$

The eigenvalue decomposition $\underline{\mathbf{A}} = \underline{\mathbf{V}}\ \underline{\Lambda}\ \underline{\mathbf{V}}^H$ leads to a similar form as the one proposed in [2]

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

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 λ_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 $\omega^2 = \lambda_i$.

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

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

with

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

and

$$Q(\omega) = \sum_i \frac{q_i}{\omega - \omega_i} \tag{4.7}$$

in the barycentric representation.

4.2 MINIMAL RATIONAL INTERPOLATION

In the following, I denote with

$$\langle u, v \rangle_M = \mathbf{u}^H \underline{\mathbf{M}} \mathbf{v} \approx \int_\Omega uv \tag{4.8}$$

the finite element approximation of the inner product in $L_2(\Omega)$. $\mathbf{u}$ and $\mathbf{v}$ are the vectors collecting the vertex values for all degrees of freedom, while $\underline{\mathbf{M}}$ is the representation matrix of the inner product in the vertex basis. Similarly, let

$$\|u\|_M = \sqrt{\langle u, u \rangle} \approx \|u\|_{L_2(\omega)} \tag{4.9}$$

For completeness, I state the strategy for numerically computing the Minimal Rational Interpolation (MRI) for a collection of snapshots sampled from the target $\mathbf{u}$ [3] in Algorithm 1. The heart of the algorithm consists in computing the Singular Value Decomposition (SVD) of the so-called Gramian matrix, and using the last left-singular vector to build the surrogate.

Algorithm 1 Minimal rational interpolation

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

4.3 GREEDY MINIMAL RATIONAL INTERPOLATION

A question that arises from the previous section is what the ideal choice of support points $\omega_1, \ldots, \omega_S$ is: How many support points are required to achieve a good enough approximation and how should the supports be distributed when given a target domain. The Greedy Minimal Rational Interpolation (gMRI) algorithm [1] tackles both questions simultaneously.

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 Ω_{test} a subset is chosen (usually the smallest and largest element), and (3.4) is solved for these two initial supports. Using these solutions, a rational surrogate is built with MRI (Algorithm 1). Motivated by the expression for the upper bound on the residual norm demonstrated in [10], new support points are chosen as the minimizers of the denominator polynomial $Q(\omega)$ and added to the set of supports. Support points are iteratively added until the relative error norm drops below a certain tolerance.

The gMRI algorithm can be found in Algorithm 2.

Algorithm 2 Greedy minimal rational interpolation

Require: $\tau > 0$ ▷ Relative L_2-error tolerance
Require: $\Omega_{\text{test}} = \{\omega_i\}_{i=1}^{M}$ ▷ Set of candidate support points
Require: $a_\omega(u, v) = L(v)$ ▷ Finite element formulation of the problem
  Choose $\omega_1, \ldots, \omega_t \in \Omega_{\text{test}}$ ▷ Usually the smallest and largest element
  Remove $\omega_1, \ldots, \omega_t$ from Ω_{test}
  Solve $a_{\omega_i}(u_i, v) = L(v)$ for $i \in \{1, \ldots, t\}$
  Build surrogate $\tilde{u}_t = P_t(\omega)/Q_t(\omega)$ using the solutions $u_1, \ldots, u_t$
  while $\Omega_{\text{test}} \neq \emptyset$ **do**
    $\omega_{t+1} \leftarrow \text{argmin}_{\omega \in \Omega_{\text{test}}} |Q_t(\omega)|$
    Solve $a_{\omega_{t+1}}(u_{t+1}, v) = L(v)$
    Build surrogate $\tilde{u}_{t+1} = P_{t+1}(\omega)/Q_{t+1}(\omega)$ using the solutions $u_1, \ldots, u_{t+1}$
    if $\|u_{t+1}(\omega_{t+1}) - \tilde{u}_{t+1}(\omega_{t+1})\|_M / \|u_{t+1}(\omega_{t+1})\|_M < \tau$ **then return**
    end if
    $t \leftarrow t + 1$
  end while

4.4 Properties of rational interpolants in barycentric coordinates

The rational surrogate $\tilde{\mathbf{u}}$ obtained with Algorithm 1 can be rewritten as

$$\tilde{\mathbf{u}}(\omega) = \sum_{j=1}^{S} \prod_{\substack{i=0 \\ i \neq j}}^{S} (\omega - \omega_i) q_j \mathbf{u}(\omega_j) / \sum_{j=1}^{S} \prod_{\substack{i=0 \\ i \neq j}}^{S} (\omega - \omega_i) q_j \tag{4.10}$$

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

4.5 Finding roots of rational functions

If the rational surrogate $\tilde{\mathbf{u}}$ is evaluated in a zero ω^* of the denominator $Q(\omega^*) = 0$, we observe a pole, provided $P(\omega^*)$ does not also vanish in that frequency. ω^* 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 a rational function of the form (**??**) can be elegantly converted to an eigenvalue problem [7]:

Define

$$\nu_i = (\omega - \omega_i)^{-1} \tag{4.11}$$

We want to find ω, such that

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

This can be equivalently expressed as the generalized eigenvalue problem

$$\underline{\mathbf{A}}\mathbf{u} = \omega \underline{\mathbf{B}}\mathbf{u} \tag{4.13}$$

with

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

4.6 Optimization tricks for greedy minimal rational interpolation

There exist many ways of improving the efficiency and/or capability of the gMRI algorithm [11]. 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 [11] alternative is to compute the QR decomposition of the snapshot matrix $\underline{\mathbf{U}} = [\mathbf{u}(\omega_1), \dots, \mathbf{u}(\omega_S)]$. Since

$$\underline{\mathbf{G}} = \underline{\mathbf{U}}^H \underline{\mathbf{M}}\, \underline{\mathbf{U}} \tag{4.15}$$

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 $\underline{\mathbf{Q}}^H \underline{\mathbf{M}}\, \underline{\mathbf{Q}} = \underline{\mathbf{1}}$. When plugging this into (4.15) one sees

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

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

$$\underline{\mathbf{R}} = \underline{\mathbf{W}}\, \underline{\mathbf{S}}\, \underline{\mathbf{V}}^H \tag{4.17}$$

Inserting this into (4.16) results in

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

which coincides with the SVD of the Gramian matrix $\underline{\mathbf{G}}$ if we take the square root of the singular values.

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 iterations of 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 [12]. 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.

Algorithm 3 Additive Householder triangularization

Require: $U[1 \dots s, 1 \dots N]$ ▷ Next snapshot matrix
Require: $R[1 \dots S, 1 \dots S]$ ▷ Previous triangular matrix
Require: $E[1 \dots S, 1 \dots N]$ ▷ Previous orthonormal matrix
Require: $V[1 \dots S, 1 \dots N]$ ▷ Previous Householder matrix
  Extend size of R to $(S+s) \times (S+s)$
  Extend E with S orthonormal columns to $(S+s) \times N$
  Extend size of V to $(S+s) \times N$
  for $j = S+1 : S+s$ **do**
    $u = U[j]$
    for $k = 1 : j-1$ **do**
      $u \leftarrow u - 2\langle V[k,:], u\rangle_M V[k,:]$
      $R[k,j] \leftarrow \langle E[k,:], u\rangle_M$
      $u \leftarrow u - R[k,j]E[k,:]$
    end for
    $R[j,j] \leftarrow \|u\|_M$
    $\alpha \leftarrow \langle E[j,:], u\rangle_M$
    if $|\alpha| \neq 0$ **then**
      $E[j,:] \leftarrow E[j,:](-\alpha/|\alpha|)$
    end if
    $V[j,:] \leftarrow R[j,j]E[j,:] - u$
    $V[j,:] \leftarrow V[j,:] - \langle E[S+1:j], V[j,:]\rangle_M E[S+1:j,:]$
    $\sigma \leftarrow \|V[j,:]\|_M$
    if $\sigma \neq 0$ **then**
      $V[j,:] \leftarrow V[j,:]/\sigma$
    else
      $V[j,:] \leftarrow E[j,:]$
    end if
  end for

4.6.2 Stability of singular value decomposition

The stability of the build of the rational surrogate using MRI can be checked by analyzing the singular values $\sigma_1, \dots, \sigma_S$ obtained from performing the SVD. 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 [11]

$$\frac{\sigma_{S-1} - \sigma_S}{\sigma_1 - \sigma_S} \tag{4.19}$$

4.6.3 ALTERNATIVE REPRESENTATIONS OF THE SURROGATE

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

$$\mathring{\mathbf{u}}(\omega) = \sum_{j=1}^{S} \frac{q_j \mathbf{e}_j}{\omega - \omega_j} / \sum_{j=1}^{S} \frac{q_j}{\omega - \omega_j} \tag{4.20}$$

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

$$\tilde{\mathbf{u}}(\omega) = \underline{\mathbf{U}} \mathring{\mathbf{u}}(\omega) \tag{4.21}$$

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

Additionally, let

$$\hat{\mathbf{u}}(\omega) = \underline{\mathbf{R}} \mathring{\mathbf{u}}(\omega) \tag{4.22}$$

with $\underline{\mathbf{R}}$ being the triangular matrix stemming from the QR decomposition of the snapshot matrix $\underline{\mathbf{U}} = \underline{\mathbf{Q}}\, \underline{\mathbf{R}}$. The original rational surrogate can again be recovered via

$$\tilde{\mathbf{u}}(\omega) = \underline{\mathbf{Q}} \hat{\mathbf{u}}(\omega) \tag{4.23}$$

$\hat{\mathbf{u}}(\omega)$ provides us with a simplified alternative to the computation of the error in gMRI (Algorithm 2).

$$\|\mathbf{u}_{t+1}(\omega_{t+1}) - \tilde{\mathbf{u}}_{t+1}(\omega_{t+1})\|_M = \|\underline{\mathbf{Q}}\mathbf{r}_{t+1} - \underline{\mathbf{Q}}\hat{\mathbf{u}}_{t+1}(\omega_{t+1})\|_M = \|\mathbf{r}_{t+1} - \hat{\mathbf{u}}_{t+1}(\omega_{t+1})\|_M \tag{4.24}$$

where $\mathbf{r}_{t+1}$ is the last column in $\underline{\mathbf{R}}^{(t+1)}$, he triangular matrix from the $(t+1)$-th step in the additive Housholder decomposition (Algorithm 3).

$$\|\mathbf{u}(\omega)\|_M^2 \approx \|\tilde{\mathbf{u}}(\omega)\|_M^2 = \|\underline{\mathbf{U}} \mathring{\mathbf{u}}(\omega)\|_M^2 = \mathring{\mathbf{u}}(\omega)^H \underbrace{\underline{\mathbf{U}}^H \underline{\mathbf{M}} \underline{\mathbf{U}}}_{=\underline{\mathbf{G}} = \underline{\mathbf{R}}^H \underline{\mathbf{R}}} \mathring{\mathbf{u}}(\omega) = \hat{\mathbf{u}}(\omega)^H \hat{\mathbf{u}}(\omega) = \|\hat{\mathbf{u}}(\omega)\|^2 \tag{4.25}$$

Thus, computing the relative error norm can be approximated by only computing Euclidean norms.

5 EXAMPLES

5.1 TWO-DIMENSIONAL RECTANGULAR CAVITY

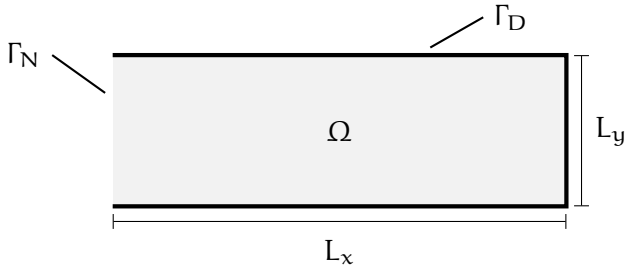
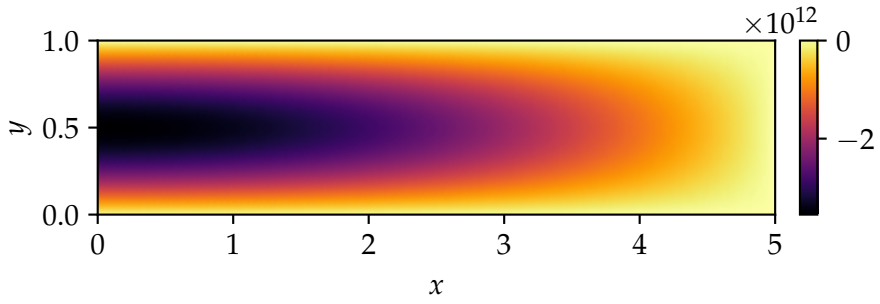
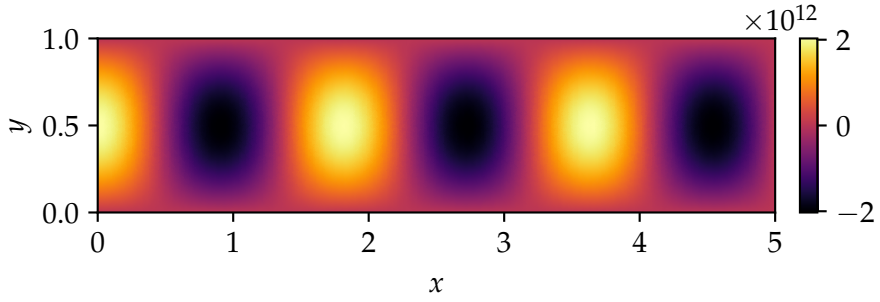


FIGURE 5.1 – TODO.



(A) First resonant frequency $\omega = 3.159$.



(B) Fifth resonant frequency $\omega = 4.675$.

FIGURE 5.2 – Caption.

For $\|\mathbf{u}\|^2_{L_2(\Omega)} = \int_\Omega \|\mathbf{u}\|^2$
Analytical eigenfrequencies

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

Numerical eigenfrequencies, solve generalized (symmetric) eigenvalue problem

$$\underline{\mathbf{K}}\mathbf{u} = \omega^2\underline{\mathbf{M}}\mathbf{u} \tag{5.2}$$

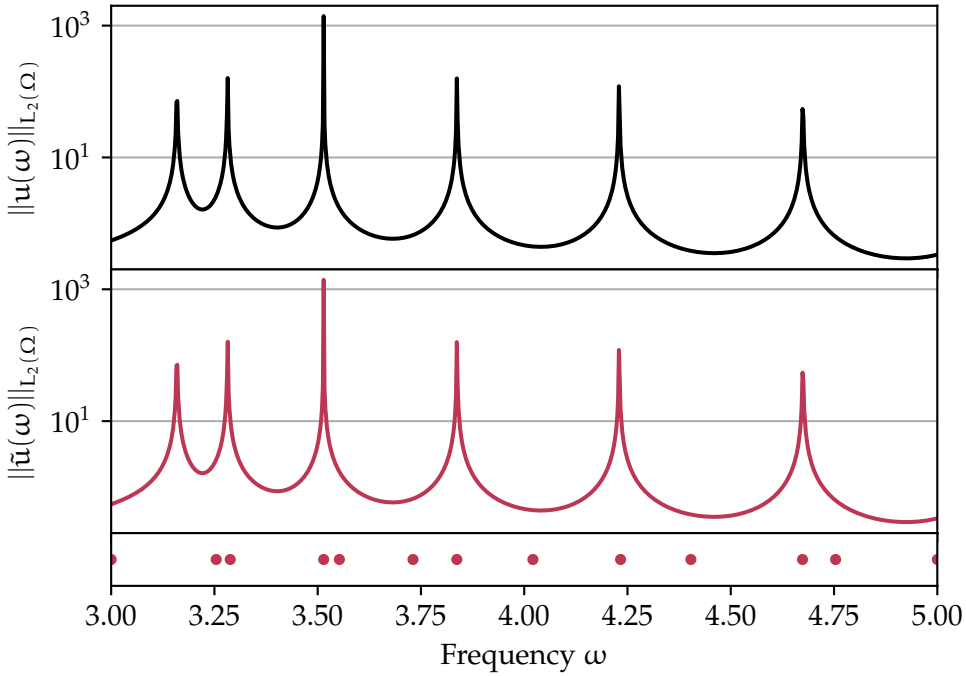


FIGURE 5.3 – Caption.

TABLE 5.1 – Comparison eigsh and gMRI.

	eigsh		gMRI	
DOF	Δ	t	Δ	t
713	1.950×10^{-2}	25.9 ± 1.1 ms	1.950×10^{-2}	61.9 ± 3.6 ms
7412	1.826×10^{-3}	199.0 ± 9.9 ms	1.827×10^{-3}	410 ± 16.8 ms
74722	1.817×10^{-4}	3.5 ± 0.1 s	1.820×10^{-4}	5.2 ± 0.2 s
745513	1.811×10^{-5}	75.0 ± 1.6 s	1.846×10^{-5}	104.0 ± 1.1 s

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

$$\underline{\mathbf{K}}\mathbf{u}_j = \omega_j^2 \underline{\mathbf{M}}\mathbf{u}_j \tag{5.3}$$

Adding a source term $\mathbf{f}$

$$\underline{\mathbf{K}}\mathbf{u} - \omega^2 \underline{\mathbf{M}}\mathbf{u} = \mathbf{f} \tag{5.4}$$

If $\mathbf{u}$ is expressed in terms of the basis $\{\mathbf{u}_j\}_j$, i.e. $\mathbf{u} = \sum_j \alpha_j \mathbf{u}_j$

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

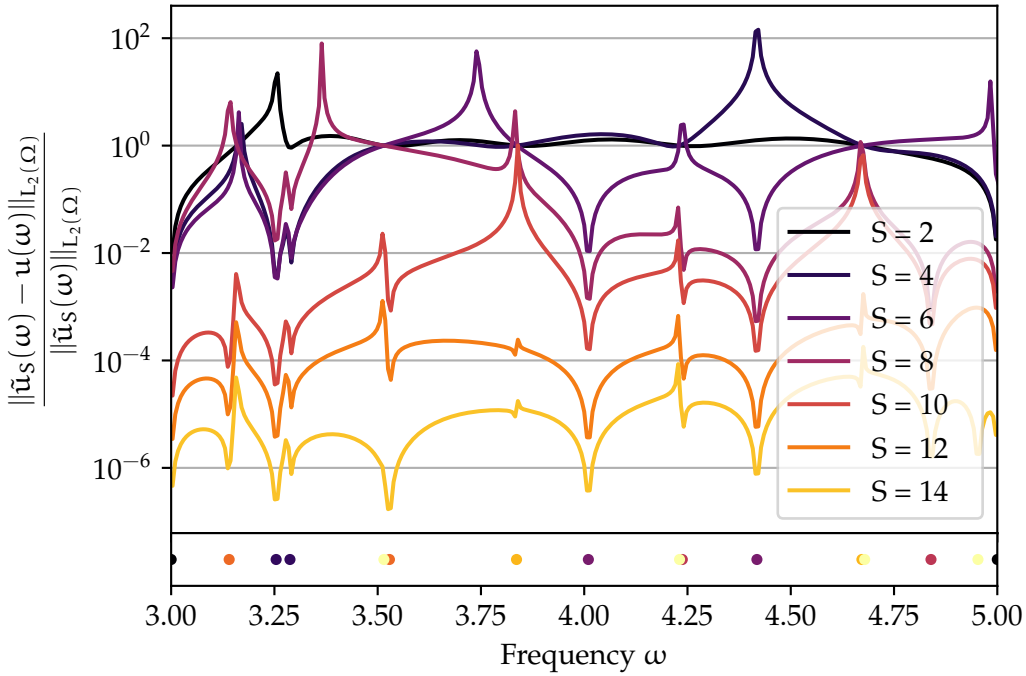


FIGURE 5.4 – Caption.

Using (5.3)

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

from which we can take the scalar product with $\mathbf{u}_j^H$ to obtain

$$\alpha_j = \frac{\mathbf{u}_j^H \mathbf{f}}{\omega_j^2 - \omega^2} \tag{5.7}$$

If $\mathbf{u}_j^H \mathbf{f} = 0$, then the resonant mode at ω_j is suppressed (fine with MRI, but eigsh will detect a resonant mode).

For $\|\mathbf{u}\|^2_{L_2(\Gamma)} = \int_\Gamma \|\mathbf{u}\|^2$

5.2 IMPERFECTLY CONDUCTING BOUNDARIES

Numerical eigenfrequencies, solve

$$(\underline{\mathbf{K}} - i\omega\underline{\mathbf{I}} - \omega^2\underline{\mathbf{M}})\mathbf{u} = 0 \tag{5.8}$$

Define $\mathbf{v} = \omega\mathbf{u}$, so that we may write this as the generalized eigenvalue problem

$$\begin{bmatrix} \underline{\mathbf{1}} & \\ \underline{\mathbf{K}} & -i\underline{\mathbf{I}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \omega \begin{bmatrix} \underline{\mathbf{1}} & \\ & \underline{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \tag{5.9}$$

which is, however, no longer Hermitian (LHS chosen as "diagonal" as possible).

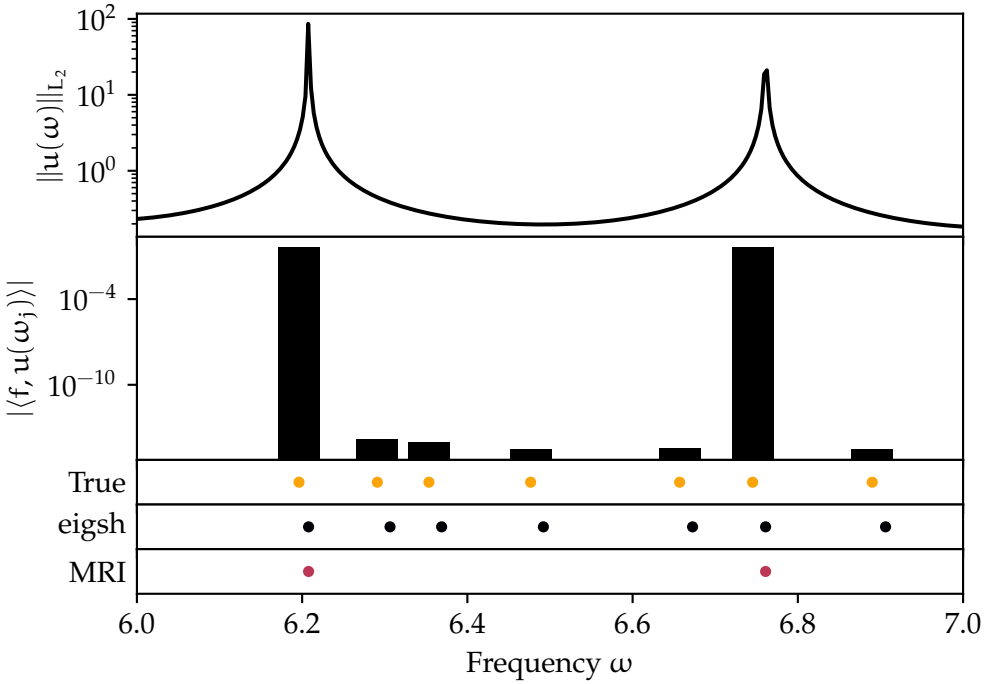


FIGURE 5.5 – Caption.

TABLE 5.2 – Comparison eigsh and gMRI.

	eigsh	gMRI
DOF	t	t
713	65.1 ± 2.48 ms	78.5 ± 7.4 ms
7412	906.0 ± 115.0 ms	496.0 ± 53.8 ms
74722	20.4 ± 0.3 s	6.2 ± 0.3 s

5.3 DUAL MODE CIRCULAR WAVEGUIDE FILTER

Modeled according to [5] in the computer-aided design modeler software application FreeCAD[2]. Mesh generated using Gmsh[3]. 5 smoothing steps, element size factor of 0.2 and Delaunay 3D meshing algorithm. The mesh was refined around critical components such as the screws and irises using transfinite curves. Conversion using meshio[4] to .xml format which can be understood by FEniCS.

[1]

$$\underline{\mathbf{S}}(\omega) = \underline{\mathbf{1}} - 2\left(\underline{\mathbf{1}} + \mathrm{i}\frac{\omega}{2\pi}\sqrt{\frac{1-(\omega_c/\omega_0)^2}{1-(\omega_c/\omega)^2}}\underline{\mathbf{F}}^{\mathrm{H}}\underline{\mathbf{U}}(\omega)\right)^{-1} \tag{5.10}$$

[2]https://www.freecadweb.org/
[3]https://gmsh.info/
[4]https://github.com/nschloe/meshio

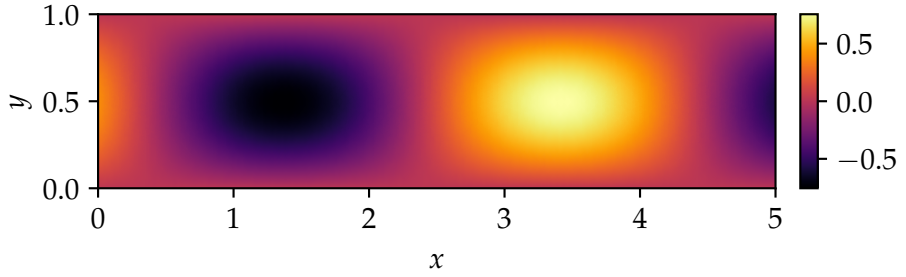


FIGURE 5.6 – $\omega = 3.5$.

with $\underline{\mathbf{F}} = [\mathbf{f}_1, \mathbf{f}_2]$ and $\underline{\mathbf{U}} = [\mathbf{u}_1, \mathbf{u}_2]$ with f_1 the term resulting when forcing from one side producing the solution $\mathbf{u}_1$, and the f_2 when forcing from the other side to produce $\mathbf{u}_2$.

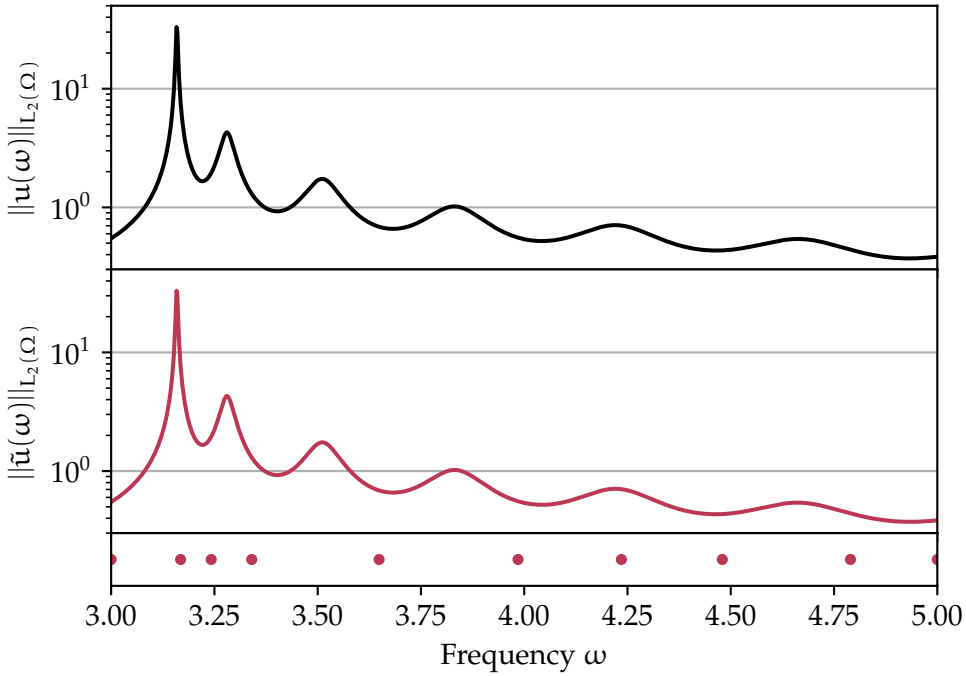


FIGURE 5.7 – Caption.

6 CONCLUSION AND OUTLOOK

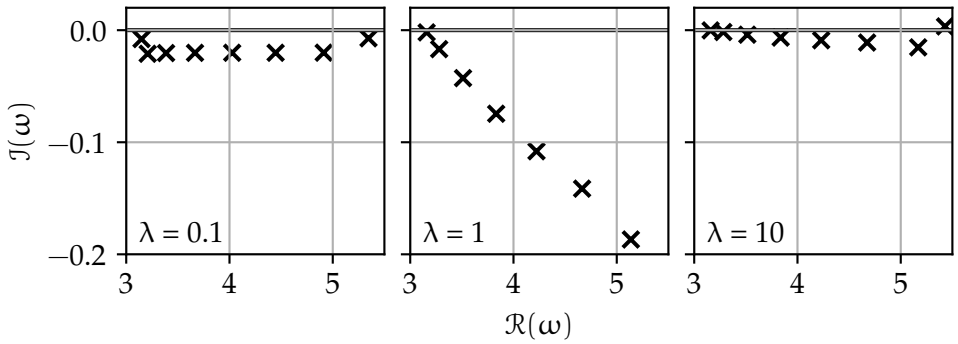


FIGURE 5.8 – Spurious resonant frequency far away from others and definitely not of interest for this problem, but indeed for rational surrogate.

REFERENCES

[1] F. Nobile D. Pradovera. 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.

[2] I. Perugia F. Bonizzoni, F. Nobile. Convergence analysis of padé approximations for helmholtz frequency response problems. *ESAIM: M2AN*, 52(4):1261 – 1284, 2018. doi: 10.1051/m2an/2017050.

[3] M. Ruggeri F. Bonzzoni, D. Pradovera. Rational-based model order reduction of helmholtz frequency response problems with adaptive finite element snapshots. doi: 10.48550/arXiv.2112.04302.

[4] A. Logg H. P. Langtangen. *Solving PDEs in Python: The FEniCS Tutorial I*. Springer, 2016. ISBN 978-3-319-52461-0. doi: 10.1007/978-3-319-52462-7.

[5] J. Zapata J. R. Montejo-Garai. Full-wave design and realization of multicoupled dual-mode circular waveguide filters. *IEEE Transactions on Microwave Theory and Techniques*, 43(6):1290 – 1291, 1995.

[6] F. Kagerer. Finite elements for maxwell's equations, 2018.

[7] G. Klein. Applications of linear barycentric rational interpolation, 2012.

[8] P. Monk. *Finite Element Methods for Maxwell's Equations*. Oxford Science Publications, 2003. ISBN 0-19-850888-3.

[9] F. Nobile. Lecture notes of the course numerical approximation of partial differential equations - i, 2017-2018.

[10] 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.

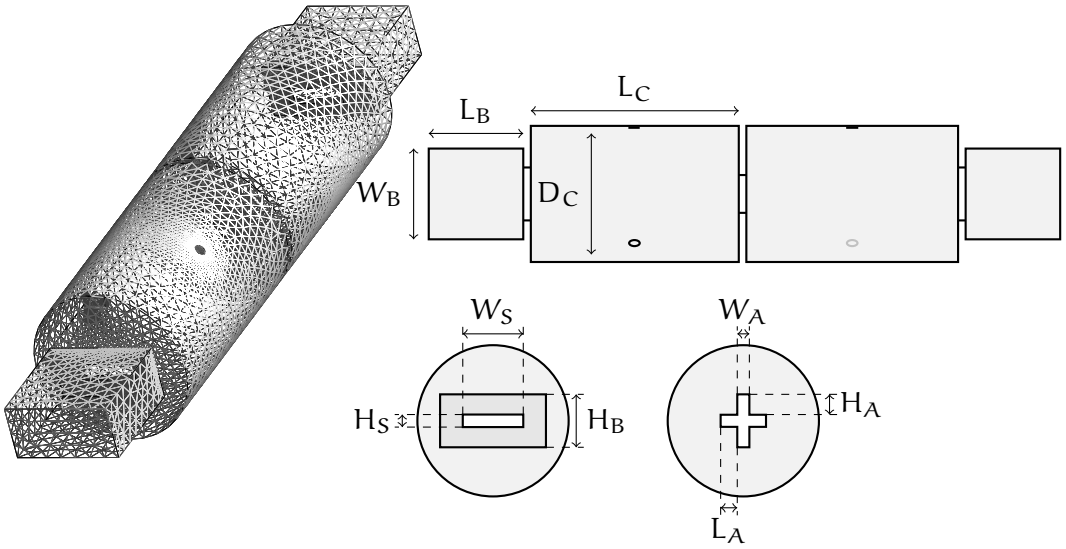


FIGURE 5.9 – Dual-mode circular waveguide filter. [5] $W_C = 43.87$ mm, $D_C = 28.0$ mm, $L_B = 43.87$ mm, $W_B = 19.05$ mm, $H_B = 9.525$ mm, $L_B = 20.0$ mm, $W_S = 10.05$ mm, $H_S = 3.0$ mm, $W_A = 2.0$ mm, $H_A = 3.375$ mm, $L_A = 2.825$ mm, thickness of all irises 2.0 mm, screws half way up the cavity horizontal tuning screws with depth 3.82 mm and coupling screws at angles $\pm 45^\circ$ with depth 3.57 mm.

[11] D. Pradovera. Model order reduction based on functional rational approximants for parametric pdes with meromorphic structure, 2021.

[12] L. N. Trefethen. Householder triangularization of a quasimatrix. *IMA Journal of Numerical Analysis*, 30(4):887–897, 2010. doi: 10.1093/imanum/drp018.

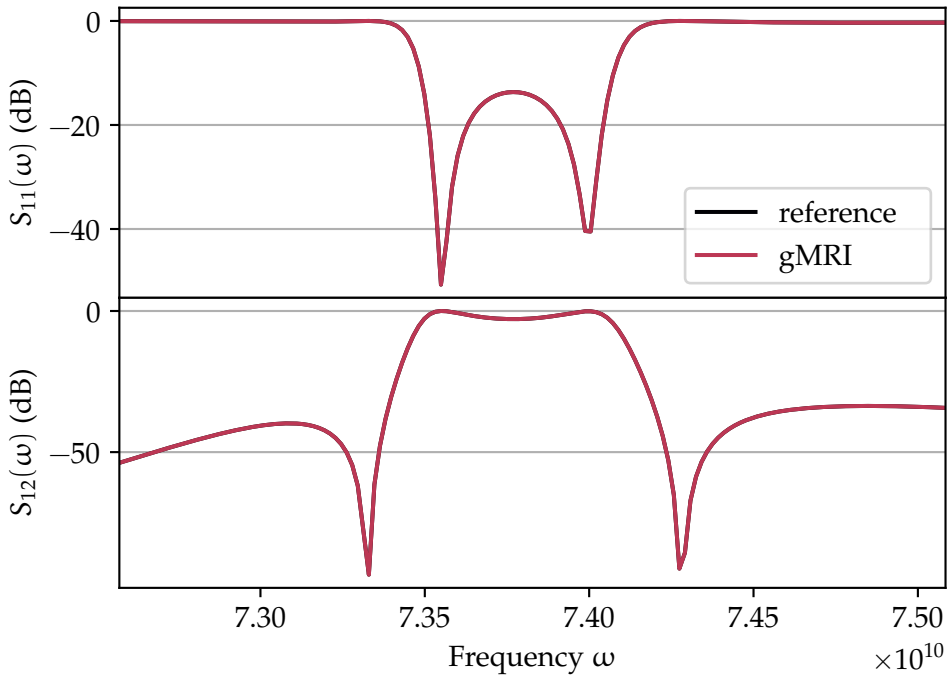


FIGURE 5.10 – Caption.

7 APPENDIX

7.1 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):

$$\int_\Omega (\nabla \times (\mu^{-1} \nabla \times \mathbf{u})) \cdot \mathbf{v} \tag{7.1}$$

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

Curl product rule

$$(\nabla \times \mathbf{a}) \cdot \mathbf{b} = \nabla \cdot (\mathbf{a} \times \mathbf{b}) + \mathbf{a} \cdot (\nabla \times \mathbf{b}) \tag{7.2}$$

where **a**, **b** are vector-value functions. The completely antisymmetric tensor ε_{ijk}, frequently referred to as the Levi-Civita tensor, may be employed to rewrite the components of the curl of a vector-function **a** as the sum

$$(\nabla \times \mathbf{a})_k = \sum_i \sum_j \varepsilon_{ijk} \partial_i u_j \tag{7.3}$$

where ∂_i denotes the partial derivative with respect to the i-th coordinate direction.

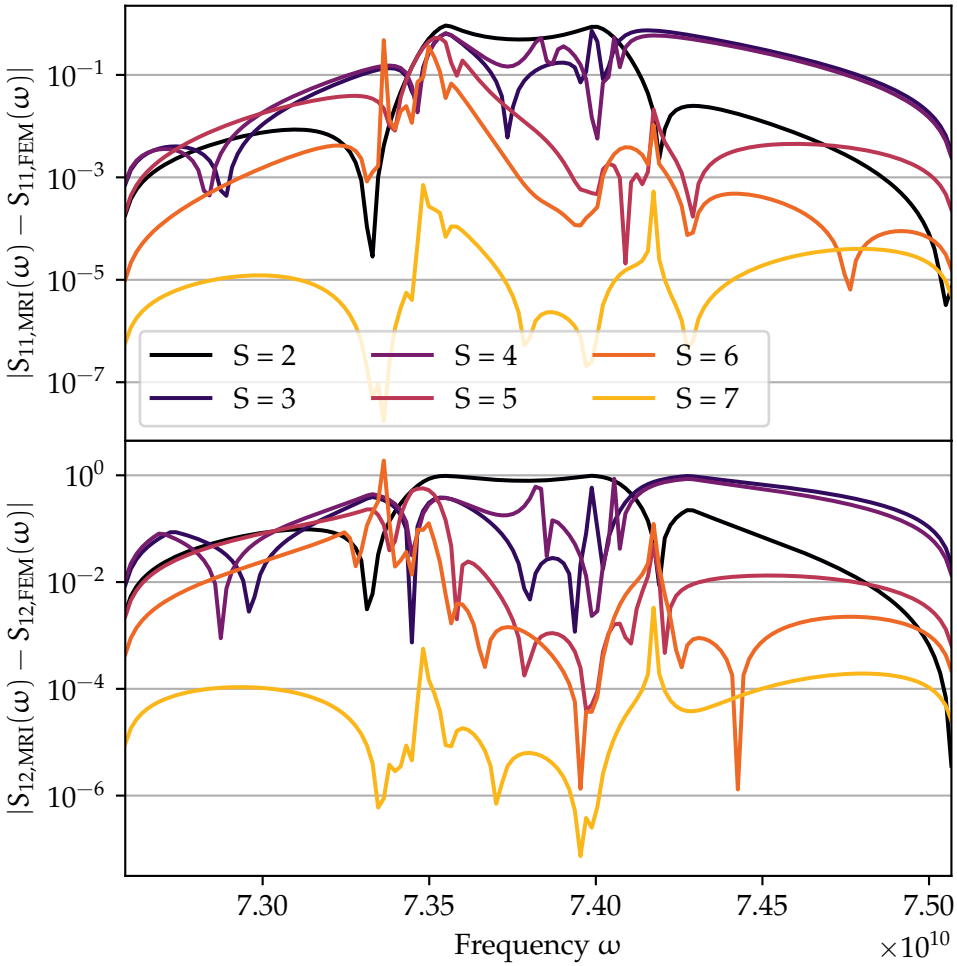


FIGURE 5.11 – Caption.

This yields

$$
\begin{aligned}
(\nabla \times \mathbf{a}) \cdot \mathbf{b} &= \sum_k (\nabla \times \mathbf{a})_k b_k \\
&= \sum_k \Big(\sum_i \sum_j \varepsilon_{ijk} \partial_i a_j\Big) b_k \\
&= \sum_k \sum_i \sum_j \partial_i(\varepsilon_{ijk} a_j b_k) - \sum_k \sum_i \sum_j a_j(\varepsilon_{ijk} \partial_i b_k) \\
&= \sum_k \sum_i \sum_j \partial_i(\varepsilon_{jki} a_j b_k) - \sum_k \sum_i \sum_j a_j((-\varepsilon_{ikj}) \partial_i b_k) \\
&= \sum_i \partial_i(\mathbf{a} \times \mathbf{b})_i + \sum_j u_j(\nabla \times \mathbf{b})_j \\
&= \nabla \cdot (\mathbf{a} \times \mathbf{b}) + \mathbf{a} \cdot (\nabla \times \mathbf{b})
\end{aligned} \tag{7.4}
$$

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 (7.2) to (7.1) together with Gauss' theorem gives

$$\begin{aligned}\int_\Omega (\nabla \times (\mu^{-1}\nabla \times \mathbf{u})) \cdot \mathbf{v} &= \int_\Omega \nabla \cdot ((\mu^{-1}\nabla \times \mathbf{u}) \times \mathbf{v}) + \int_\Omega (\mu^{-1}\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v}) \\ &= \int_{\partial\Omega} ((\mu^{-1}\nabla \times \mathbf{u}) \times \mathbf{v}) \cdot \mathbf{n} + \int_\Omega (\mu^{-1}\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v})\end{aligned} \tag{7.5}$$

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

Commutative behavior of the scalar triple product

$$(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} = -(\mathbf{a} \times \mathbf{c}) \cdot \mathbf{b} \tag{7.6}$$

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

$$\begin{aligned}(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} &= \sum_k (\sum_i \sum_j \varepsilon_{ijk} a_i b_j) c_k \\ &= \sum_j (\sum_i \sum_k (-\varepsilon_{ikj}) a_i c_k) b_j \\ &= -(\mathbf{a} \times \mathbf{c}) \cdot \mathbf{b}\end{aligned} \tag{7.7}$$

The boundary integral becomes

$$\int_{\partial\Omega} ((\mu^{-1}\nabla \times \mathbf{u}) \times \mathbf{v}) \cdot \mathbf{n} = -\int_{\partial\Omega} ((\mu^{-1}\nabla \times \mathbf{u}) \times \mathbf{n}) \cdot \mathbf{v} \tag{7.8}$$

This concludes the short derivation, because now (7.1) may be rewritten as

$$-\int_{\partial\Omega} ((\mu^{-1}\nabla \times \mathbf{u}) \times \mathbf{v}) \cdot \mathbf{n} + \int_\Omega (\mu^{-1}\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v}) \tag{7.9}$$

I would also like to demonstrate that

Curl product rule

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

$$(\mathbf{n} \times \mathbf{u}) \times \mathbf{n} = \mathbf{u} \tag{7.10}$$

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

$$\sum_i \varepsilon_{jki}\varepsilon_{lmi} = \delta_{jl}\delta_{km} - \delta_{jm}\delta_{kl} \tag{7.11}$$

Furthermore, I make use of the invariance of the tensor under cyclic permutation of the indices to obtain

$$\begin{aligned}
[(\mathbf{n}\times\mathbf{u})\times\mathbf{n}]_k &= \sum_i\sum_j \varepsilon_{ijk}(\mathbf{n}\times\mathbf{u})_i n_j \\
&= \sum_i\sum_j \varepsilon_{ijk}\sum_l\sum_m \varepsilon_{lmi} n_l u_m n_j \\
&= \sum_i\sum_j\sum_l\sum_m \varepsilon_{jki}\varepsilon_{lmi} n_l u_m n_j \\
&= \sum_j\sum_l\sum_m (\delta_{jl}\delta_{km}-\delta_{jm}\delta_{kl}) n_l u_m n_j \\
&= \sum_j n_j u_k n_j - \sum_j n_k u_j n_j \\
&= \|\mathbf{n}\|^2 u_k - (\mathbf{u}\cdot\mathbf{n}) n_k \\
&= u_k
\end{aligned} \tag{7.12}$$

which concludes the component-wise proof.