BACHELORARBEIT

Discrete filters for a Krylov eigenvalue solver

ausgeführt am

Institut für
Analysis und Scientific Computing
TU Wien

unter der Anleitung von

Associate Prof. Dipl.-Math. Dr.rer.nat. Lothar Nannen
Univ.Ass. Dipl.-Ing. Dr.techn. Markus Wess

durch

Michał Trojanowski
Matrikelnummer: 12108865

Wien, am 5. Juni 2024

Acknowledgement

I am deeply grateful to my supervisors, Prof. Dr. Lothar Nannen and Dr. Markus Wess, for their guidance and continuous support throughout the writing of this thesis. My appreciation also goes out to my parents and friends for their encouragement and support during my studies.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 5. Juni 2024

Michał Trojanowski

Contents

1 Introduction

In this thesis, we consider the eigenvalue problem of the negative Laplace operator on a 2- or 3-dimensional domain with Neumann boundary conditions. We compute the eigenpairs (ω^2, u) using a Krylov eigenvalue solver based on time domain solutions, following the approach introduced in [NW24]. This problem is significant in the theory of wave equations and physics, e.g., for the computation of acoustic resonances.

We begin with the standard finite element method: discretizing the domain, fixing the finite-dimensional discrete solution space, and reformulating the problem to its weak form. This leads to a matrix eigenvalue problem to find eigenpairs (ω^2, v), such that $Sv = \omega^2 Mv$, where the matrices S and M are sparse, symmetric, and (semi-)positive definite.

Elementary numerical solvers for eigenvalue problems struggle when the dimension of the matrix becomes large. Therefore, instead of using a direct solver, we perform a Krylov iteration, project the large-scale problem onto a lower-dimensional Krylov space, and solve this reduced problem using a direct solver with lower computational costs. We seek eigenvalues in a specific region of interest, so we want the Krylov space to primarily cover the eigenspaces corresponding to the sought eigenvalues.

As an operator for the iteration, we could use the shift-and-inverse matrix, but this would require computing the inverse of a large matrix at each step, which is out of reach. A remedy is to introduce an operator Π_α – a weighted time-integral of the solution to the semi-discrete wave equation, and its discrete equivalent C, which uses a quadrature rule instead of the integral and the finite difference method instead of the analytical solution.

The construction of this operator requires a suitable weight function α or a weight vector $\vec{\alpha}$. Crucial to the method is a filter function β_α and its discrete counterpart $\tilde{\beta}_\alpha$, which, depending on the weight function α, quantifies the correspondence between eigenvalues of the original problem and those of the operator C.

In this thesis, we discuss several approaches to obtaining weights that yield an appropriate filter function. We start with a method using the inverse Fourier transform. As it turns out, the discrete filter function is a polynomial in ω^2. Thus, it is natural to reduce the problem to a polynomial collocation or a least squares minimization. In the third approach, we minimize the L^2 norm of the residuum between the obtained filter function $\tilde{\beta}_{\vec{\alpha}}$ and the desired goal function.

The structure of this paper is as follows. First, we state the problem, discretize it in space, and review elementary numerical methods for matrix eigenvalue problems. Following [NW24], we present the Krylov eigenvalue solver. As a result of Chapter 2, we formulate an implementable algorithm for solving the eigenvalue problem of the negative Laplace operator. In the subsequent part, we discuss the choice of the weight function and consider different approaches to obtain it. Then, in Chapter 4, we solve the original problem on a simple 2-dimensional rectangular domain to demonstrate the impact of the filter function on the convergence of the eigenvalues and the reliability of the algorithm.

2 A Krylov eigenvalue solver based on filtered time domain solutions

In this chapter, we introduce the concept of a Krylov eigenvalue solver for computing the eigenvalues of the negative Laplace operator. By reformulating the problem to its weak form and selecting a finite solution space, we find that our problem is equivalent to a matrix eigenvalue problem for some high-dimensional matrix. The core idea of the Krylov eigenvalue solver is to construct a matrix C that has the same eigenspaces as the original problem but different eigenvalues. Projection onto a lower-dimensional Krylov space generated by this matrix significantly reduces computational costs. We construct the operator C by integrating in time a solution to the wave equation, multiplied by a weight function. Proper selection of the weight function allows us to find the eigenvalues of the original problem in the desired region. This chapter is based on [NW24].

Definition 2.1 (Eigenvalue problem of the negative Laplace operator)**.** *Let $\Omega \subset \mathbb{R}^d$ for $d = 2, 3$ be a bounded domain with a Lipschitz boundary. We seek eigenvalues $\omega^2 \in \mathbb{R}_+$ and eigenfunctions $u \in H^1(\Omega)\backslash\{0\}$ of the negative Laplace operator with Neumann boundary conditions:*

$$\begin{aligned} -\Delta u &= \omega^2 u \quad \text{in } \Omega, \\ \frac{\partial u}{\partial \nu} &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{2.1}$$

Here $\frac{\partial}{\partial \nu}$ denotes the normal derivative, and H^1 is the Sobolev space.

Now we discretize the problem in space, fixing a partition $\mathcal{T}$ of Ω. We introduce a discrete solution space as a finite-dimensional space of piecewise polynomials on Ω.

Definition 2.2 (Discrete solution space)**.** *Let $\mathbb{P}_p$ denote the space of polynomials up to degree $p \in \mathbb{N}$. We define the discrete solution space as*

$$V_h := \left\{ v \in H^1(\Omega) : \quad \forall T \in \mathcal{T},\, v|_T \in \mathbb{P}_p \right\}$$

with finite dimension $N := \dim(V_h)$.

Using Gauss's theorem, we can formulate both the weak and discrete forms of problem (2.1).

Definition 2.3 (Weak formulation of the eigenvalue problem)**.** *Let Ω be a bounded Lipschitz domain, as defined in Definition 2.1, and V_h be the discrete solution space. We seek eigenvalues $\omega_h^2 \in \mathbb{R}_+$ corresponding to discrete eigenfunctions $u \in V_h\backslash\{0\}$, such that for all test functions $\varphi \in V_h$, the following holds:*

$$\int_\Omega \nabla u \cdot \nabla \varphi \, dx = \omega_h^2 \int_\Omega u\varphi \, dx. \tag{2.2}$$

To simplify notation in subsequent sections, we omit the index h for discrete eigenfunctions and eigenvalues. We now demonstrate that the discrete eigenvalue problem (2.2) reduces to a matrix eigenvalue problem, since the solution space is finite-dimensional.

Definition 2.4. *Let V_h be an N-dimensional solution space with basis $(\varphi_1, \dots, \varphi_N)$. We define matrices $S := (s_{ij})_{i,j=1}^N$ and $M := (m_{ij})_{i,j=1}^N$ as follows:*

$$s_{ij} := \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, dx \quad \text{and} \quad m_{ij} := \int_\Omega \varphi_i\varphi_j \, dx.$$

Lemma 2.5. *Equivalent to the problem stated in Definition 2.3 is the eigenvalue problem to find non-trivial $v \in \mathbb{R}^N$ and $\omega^2 \in \mathbb{R}_+$ such that:*

$$Sv = \omega^2 Mv \tag{2.3}$$

where $S, M \in \mathbb{R}^{N\times N}$ are defined in Definition 2.4.

Proof. Since $(\varphi_1, \dots, \varphi_N)$ forms a basis of V_h, it suffices, if (2.2) holds for all basis functions. By representing u with its coordinate vector $v \in \mathbb{R}^N$, we can substitute $u = (\varphi_1, \dots, \varphi_N)v$. Thus, we obtain:

$$\begin{aligned}
\forall j = 1, \dots, N : \int_\Omega (\nabla\varphi_1, \dots, \nabla\varphi_N)v \cdot \nabla\varphi_j \, dx = \omega^2 \int_\Omega (\varphi_1, \dots, \varphi_N)v\varphi_j \, dx \quad &\iff \\
\forall j = 1, \dots, N : \int_\Omega (\nabla\varphi_j \cdot \nabla\varphi_1, \dots, \nabla\varphi_j \cdot \nabla\varphi_N)v \, dx = \omega^2 \int_\Omega (\varphi_j\varphi_1, \dots, \varphi_j\varphi_N)v \, dx \quad &\iff \\
\forall j = 1, \dots, N : (s_{j1}, \dots, s_{jN})v = \omega^2 (m_{j1}, \dots, m_{jN})v, &
\end{aligned}$$

which is equivalent to (2.3). □

Remark 2.6. *Matrices S and M from Definition 2.4 are self-adjoint. Furthermore, matrix S is positive semi-definite and M is positive-definite.*

2.1 Elementary eigenvalue solvers for matrices

Since we have approximated the eigenvalue problem of the negative Laplacian operator (see Definition 2.1) by a matrix eigenvalue problem, we recall elementary numerical algorithms to solve such problems. For proofs of the convergence of these algorithms, we refer to [Nan23].

The first algorithm, under assumptions of Theorem 2.7, provides an approximation of an eigenvector corresponding to the eigenvalue with the largest absolute value among the eigenvalues of the matrix.

Algorithm 1 Power iteration

Input: $A \in \mathbb{R}^{N\times N}$, start vector $v^{(0)} \in \mathbb{R}^N \backslash \{0\}$
for $i = 1, 2, \dots$ **do**
  $v \leftarrow Av^{(i-1)}$
  $v^{(i)} \leftarrow \frac{v}{\|v\|_2}$
end for
Output: $v^{(i)}$ – approximation of an eigenvector to the eigenvalue with the largest absolute value (under assumptions of Theorem 2.7).

Theorem 2.7. *Let $A \in \mathbb{R}^{N\times N}$ be a diagonalizable matrix with eigenvalues $\mu_1, \dots, \mu_N$, such that $|\mu_1| > |\mu_2| \geqslant |\mu_j|$ for all $j = 3, \dots, N$. Let $(v_1, \dots, v_N)$ denote a basis of $\mathbb{R}^N$, such that for all $j = 1, \dots N$, v_j is a normed eigenvector to eigenvalue μ_j. Let $v = \sum_{j=1}^{N} c_j v_j \in \mathbb{R}^N$ be a start vector, such that $c_1 \neq 0$. Then for all $i \in \mathbb{N}$, holds the error estimation for the eigenspace:*

$$\left\| v^{(i)} - \frac{\mu_1^i c_1}{|\mu_1^i c_1|} v_1 \right\|_2 = \mathcal{O}\left(\left| \frac{\mu_2}{\mu_1} \right|^i \right) \text{ for } i \to \infty.$$

Furthermore, holds the error estimation for the convergence of the eigenvalue. Let $\mu^{(i)} := (Ax^{(i)})_k / x_k^{(i)}$ for some $k = 1, \dots, N$. It holds:

$$|\mu_1 - \mu^{(i)}| = \mathcal{O}\left(\left| \frac{\mu_2}{\mu_1} \right|^i \right) \text{ for } i \to \infty.$$

Proof. We refer to [Nan23, p. 116]. □

The second algorithm, under assumptions of Theorem 2.8, enables us to compute the entire spectrum of a matrix.

Algorithm 2 QR Algorithm

Input: $A \in \mathbb{R}^{N\times N}$
$A^{(0)} \leftarrow A$
for $i = 1, 2, \dots$ **do**
  compute QR decomposition: $Q^{(i)} R^{(i)} = A^{(i-1)}$
  $A^{(i)} \leftarrow R^{(i)} Q^{(i)}$
end for
Output: $A^{(i)}$ – approximation of an upper triangular matrix with eigenvalues of A on the diagonal (under assumptions of Theorem 2.8).

Theorem 2.8. *Let $A \in \mathbb{R}^{N\times N}$ be a diagonalizable matrix with pairwise different absolute values of eigenvalues: $\mu_1 > \mu_2 > \dots > \mu_N$. Let $\Lambda^{(i)} := \left(A_{11}^{(i)}, \dots, A_{NN}^{(i)} \right)$. Then $\Lambda^{(i)}$ converges linearly towards $(\mu_1, \dots, \mu_N)$ as $i \to \infty$.*

Proof. We refer to [Nan23, p. 120]. □

Since our goal is to find the eigenvalues of the negative Laplace operator, we could use the QR algorithm to obtain all eigenvalues of the discrete problem. However, this is not feasible due to the high computational costs of the QR algorithm. We focus on situations where the problem at the discrete level is high-dimensional. The QR algorithm requires the computation of a QR decomposition of an $N \times N$ matrix in each iteration, which has cubic computational complexity. Therefore, this method is out of reach for our problem. Instead, we must deploy a method that significantly reduces the problem's size before employing a direct solver.

2.2 A Krylov eigenvalue solver

Now, we need to make an assumption that there exists a matrix $C \in \mathbb{R}^{N\times N}$ such that its eigenvectors are linear combinations of eigenvectors of the discrete problem (2.3).

Definition 2.9. *Let $C \in \mathbb{R}^{N\times N}$ be a matrix such that there exists an eigenvalue $\mu \in \mathbb{R}$ corresponding to the eigenvector $w \in \mathbb{R}^N$, i.e., $Cw = \mu w$, if and only if w is an eigenvector or a linear combination of eigenvectors to some eigenvalue ω^2 in the discrete problem (2.3).*

We obtain properties of the matrix C that will be needed. The exact choice and construction of this matrix will be discussed later. We recall the definition of a Krylov space.

Definition 2.10 (Krylov space)**.** *Let $C \in \mathbb{R}^{N\times N}$ be a matrix and $r \in \mathbb{R}^N$ be a normalized start vector, i.e., $\|r\|_2 = 1$. For $m \in \mathbb{N}$, we define the Krylov space as a subspace of $\mathbb{R}^N$:*

$$\mathcal{K}_m(C; r) := \operatorname{span}\left\{r, Cr, \ldots, C^{m-1}r\right\}.$$

In this thesis, we focus on situations where N is large. Our goal is to project the N-dimensional eigenvalue problem (2.3) onto an m-dimensional Krylov space, using an orthonormal basis of this space. Typically, we choose m much smaller than N, so that this problem is solvable with low computational costs using a direct solver.

Let $\mathcal{K}_m(C; r)$ be an m-dimensional Krylov space generated by matrix $C \in \mathbb{R}^{N\times N}$ and normalized start vector $r \in \mathbb{R}^N$. We obtain an orthonormal basis $(b_0, \ldots, b_{m-1})$ of the Krylov space using Gram-Schmidt orthonormalization:

$$b_0 := r \quad \text{and} \quad \widetilde{b}_j := Cb_{j-1} - \sum_{i=0}^{j-1} (b_i^T C b_{j-1}) b_i, \quad b_j := \frac{\tilde{b}_{j-1}}{\|\tilde{b}_{j-1}\|_2} \quad \text{for all } j = 1, \ldots, m-1.$$

Now we can project the original problem (2.3) onto the m-dimensional Krylov space.

Proposition 2.11 (Eigenvalue problem on a Krylov space)**.** *Let $B_m = (b_0, \ldots, b_{m-1}) \in \mathbb{R}^{N\times m}$ be an orthonormal basis of an m-dimensional Krylov space $\mathcal{K}_m(C; r)$. The eigenvalue problem on the Krylov space is to find eigenvalues $\omega_m^2 \in \mathbb{R}_+$ and eigenvectors $v_m \in \mathbb{R}^m$, such that:*

$$B_m^T S B_m v_m = \omega_m^2 B_m^T M B_m v_m. \tag{2.4}$$

Since matrices S and M are symmetric, Krylov iteration leads to convergence of eigenspaces of (2.4) towards the eigenspace of C corresponding to the eigenvalue $\mu_{\max}$ with the largest absolute value. Obviously, projected eigenvalues ω_m^2 converge towards the eigenvalue ω^2 of the original problem (2.3) corresponding to $\mu_{\max}$.

To sum up the core idea of this section, we can explicitly formulate an (not yet implementable) algorithm based on Krylov spaces to compute eigenvalues ω^2.

Algorithm 3 Krylov eigenvalue solver

Input: matrix C, start vector r, M^{-1}, S, m dimension of the Krylov space
$b_0 \leftarrow r$
for $k = 1, \dots, m-1$ **do**
  $b_k \leftarrow Cb_{k-1}$ ▷ Krylov step
  $b_k \leftarrow b_k - \sum_{i=0}^{k-1}(b_i^T b_k)b_i$ ▷ Gram-Schmidt orthogonalization
  $b_k \leftarrow b_k / \|b_k\|_2$ ▷ Normalization
end for
$B_m \leftarrow (b_0, \dots, b_{m-1})$ ▷ Projection matrix
$A \leftarrow B_m^T M^{-1} S B_m$
solve $Av = \omega_m^2 v$ with a direct solver

Our goal is to compute eigenvalues ω^2 of (2.3) in a given region of interest $\left[\omega_{\min}^2, \omega_{\max}^2\right]$. Therefore, we need matrix C to fulfill the following four conditions.

- C satisfies the conditions in Definition 2.9.
- Eigenvalues μ corresponding to eigenspaces of eigenvalues ω^2 in the region of interest have large absolute value.
- Eigenvalues μ corresponding to eigenspaces of eigenvalues ω^2, which are not sought, are close to 0.
- Each Krylov step $r \mapsto Cr$ can be computed with low cost.

In other words, the use of matrix C filters sought eigenvalues from among all eigenvalues of the original problem (2.3) via common eigenspaces.

Remark 2.12. *Shift-and-inverse matrix:*

$$C_\rho := (S - \rho M)^{-1} M$$

for some $\rho \in \mathbb{R}$ has eigenvalue $\mu = (\omega^2 - \rho)^{-1}$ corresponding to the eigenspace of the eigenvalue ω^2 of the original problem (2.3). Therefore, C_ρ with $\rho = (\omega_{\min}^2 + \omega_{\max}^2)/2$ would be a possible choice of matrix C. Nevertheless, it requires the inverse of an $N \times N$ matrix $(S - \rho M)$, which for large N is not efficiently computable.

Proof. Let $(\mu, w) \in \mathbb{R} \times \mathbb{R}^N$ be an eigenpair of C_ρ. We note that $\mu \neq 0$, since matrices

$(S-\rho M)^{-1}$ and M have full rank. We have:

$$\begin{aligned} C_\rho w &= \mu w && \iff \\ (S-\rho M)^{-1} M w &= \mu w && \iff \\ \frac{1}{\mu} M w &= (S-\rho M) w && \iff \\ S w &= \underbrace{\left(\rho + \frac{1}{\mu}\right)}_{=\omega^2} M w. \end{aligned}$$

Therefore ω^2 is an eigenvalue of $M^{-1}S$ to the eigenvector w. Conversely, if ω^2 is an eigenvalue of $M^{-1}S$, then $\mu = (\omega^2 - \rho)^{-1}$ is an eigenvalue of C_ρ to the same eigenvector. Obviously, $|\mu|$ takes on the largest values if $\omega^2 \approx \rho$. □

2.3 Filtered time domain solutions

To construct an appropriate finite-dimensional operator to replace the role of the matrix C from the previous section, we proceed as follows. We consider the homogeneous wave equation with some initial condition $u_0 = (\varphi_1, \dots, \varphi_N)r$ projected onto our discrete solution space V_h (see Definition 2.2). We discretize the problem in space and formulate its weak form. The evaluation of our operator C at the point r is a time-integral of the solution to this semi-discrete problem.

We consider the homogeneous 2- or 3-dimensional wave equation in $\Omega \times (0,\infty)$:

$$\begin{aligned} \ddot{u} = \Delta u \quad \text{in } \Omega \times (0,\infty), \quad \frac{\partial u}{\partial \nu} = 0 \quad \text{in } \partial\Omega \times (0,\infty), \\ u(\cdot,0) = u_0, \quad \dot{u}(\cdot,0) = 0 \quad \text{in } \Omega, \end{aligned} \tag{2.5}$$

where u is a function $\Omega \times [0,\infty) \to \mathbb{R}$ and $\ddot{u}$ denotes the second time-derivative.

Proposition 2.13. *Let V_h be a discrete solution space with basis $(\varphi_1, \dots, \varphi_N)$, and let r denote the coordinate vector of the projection of u_0 onto V_h, i.e., $u_0 = (\varphi_1, \dots, \varphi_N)r$. Let S and M denote the matrices from Definition 2.4. Discretization in space of the problem (2.5) leads to the following linear system of ordinary differential equations:*

$$\begin{aligned} M\ddot{y}(t;r) &= -Sy(t;r) \quad \text{for all } t \in (0,\infty) \\ y(0;r) &= r, \quad \dot{y}(0;r) = 0. \end{aligned} \tag{2.6}$$

Proof. The theory of partial differential equations leads us to the ansatz $u(t,x) = \sum_{i=1}^{\infty} \kappa_i(t)\xi_i(x)$, where ξ_i are the eigenfunctions of the negative Laplace operator [Jü23, p. 122]. Since we want to solve the problem in our solution space V_h, we can assume that $u(\cdot,t) = (\varphi_1, \dots, \varphi_N)y(t)$ for some vector-valued function $y : [0,\infty) \to \mathbb{R}^N$.

We reformulate $\ddot{u} = \Delta u$ to its weak form:

$$\begin{aligned} &\forall \varphi \in V_h : \int_\Omega \ddot{u}\varphi \, dx = \int_\Omega \Delta u \varphi \, dx && \iff \\ &\forall \varphi \in V_h : \int_\Omega \ddot{u}\varphi \, dx = -\int_\Omega \nabla u \cdot \nabla \varphi \, dx. \end{aligned}$$

V_h is a finite-dimensional space, and integration and differentiation are linear operators. Therefore, the condition "for all test functions" is equivalent to the condition "for all basis test functions":

$$\forall j = 1, \ldots, N : \int_\Omega (\varphi_1, \ldots, \varphi_N)\ddot{y}\varphi_j \, dx = -\int_\Omega (\nabla\varphi_1, \ldots, \nabla\varphi_N) y \cdot \nabla\varphi_j \, dx \qquad \Longleftrightarrow$$
$$M\ddot{y} = -Sy.$$

It remains to show the equivalence of the initial conditions. For u_0, we have:

$$(\varphi_1, \ldots, \varphi_N)r = u_0 = u(\cdot, 0) = (\varphi_1, \ldots, \varphi_N)y(0),$$

so $y(0) = r$ in Ω, and similarly $\dot{y}(0) = 0$ in Ω. □

Now, we can solve the semi-discrete wave equation (2.6) to construct an integral operator.

Lemma 2.14. *The unique solution to (2.6) is:*

$$y(t; r) = \sum_{j=1}^{N} \cos(\omega_j t)(v_j^T r)v_j, \tag{2.7}$$

where $(v_j)_{j=1}^N$ is an orthonormal basis of $\mathbb{R}^N$ of eigenvectors of matrix $M^{-1}S$ with corresponding eigenvalues ω_j^2.

Proof. Because of the symmetry of matrices S and M, there exists an orthonormal basis $(v_j)_{j=1}^N$ of $\mathbb{R}^N$ consisting of eigenvectors of $M^{-1}S$. We denote corresponding eigenvalues with ω_j^2, $j = 1, \ldots, N$. Obviously, for all $j = 1, \ldots, N$ and for all $c_{1j}, c_{2j} \in \mathbb{R}$, the function $t \mapsto (c_{1j}\cos(\omega_j t) + c_{2j}\sin(\omega_j t))\, v_j$ solves the differential equation (2.6). Due to the linearity of the problem,

$$y(t) = \sum_{j=1}^{N} (c_{1j}\cos(\omega_j t) + c_{2j}\sin(\omega_j t))\, v_j$$

solves the differential equation. Initial values lead to the form (2.7). The uniqueness of the solution follows from the Picard-Lindelöf theorem. □

Now we can define an integral operator that maps the initial value of the semi-discrete wave equation (2.6) to a weighted time-integral of its unique solution. The discrete form of this operator will take over the role of matrix C. Depending on the choice of the weight function, eigenvalues of this discrete operator may fulfill the requirements that we have set for matrix C in the previous section. This will be discussed later in Chapter 3.

Definition 2.15. *Let $\alpha : [0; \infty) \to \mathbb{R}$ be a piecewise continuous function with a compact support. We define $\Pi_\alpha : \mathbb{R}^N \to \mathbb{R}^N$ as a linear operator:*

$$\Pi_\alpha r := \int_0^\infty \alpha(t) y(t; r)\, dt,$$

where $y(t; r)$ is the unique solution to (2.6) from Lemma 2.14.

The following lemma determines the correspondence between eigenvalues of Π_α and eigenvalues of the original problem (2.3).

Lemma 2.16. *Let $\beta_\alpha : [0, \infty) \to \mathbb{R}$ be a filter function defined by:*

$$\beta_\alpha(\omega) := \int_0^\infty \alpha(t) \cos(\omega t)\, dt. \tag{2.8}$$

Then the following two statements hold.

1. *If ω^2 is an eigenvalue of (2.3) corresponding to eigenvector v, then v is also an eigenvector of Π_α corresponding to eigenvalue $\beta_\alpha(\omega)$.*
2. *If λ is an eigenvalue of Π_α to eigenvector v, then there exists an eigenvalue ω^2 of (2.6) such that $\beta_\alpha(\omega) = \lambda$, and*

$$v \in \bigoplus \{U : \quad \exists \omega : U \text{ is eigenspace to eigenvalue to } \omega \text{ and } \beta_\alpha(\omega) = \lambda\}.$$

Proof. 1. Since $(v_j)_{j=1}^N$ is an orthonormal basis consisting of eigenvectors of matrix $M^{-1}S$, for every eigenvector v of (2.3) exists some $j = 1, \ldots, N$, such that $v = v_j$ with v_j from (2.7). Therefore, $y(t, v) = \cos(\omega_j t)v$ and

$$\Pi_\alpha v = \int_0^\infty \alpha(t) \cos(\omega_j t) v\, dt = \beta_\alpha(\omega) v.$$

So the first claim holds.

2. If λ is an eigenvalue of Π_α with eigenvector v, then for all $k = 1, \ldots, N$ holds:

$$\begin{aligned} 0 = (\Pi_\alpha v - \lambda v) v_k &= \int_0^\infty \alpha(t) \sum_{j=1}^N \cos(\omega_k t)(v_j^T v) \underbrace{(v_j^T v_k)}_{=\delta_{jk}}\, dt - \lambda v^T v_k \\ &= \left(\int_0^\infty \alpha(t) \cos(\omega_k t) - \lambda \right) v_k^T v = (\beta_\alpha(\omega_k) - \lambda) v_k^T v. \end{aligned}$$

If $\beta_\alpha(\omega_k) \neq \lambda$, then $v_k^T v = 0$. Therefore v belongs to the sum of eigenspaces to eigenvalues such that $\beta_\alpha(\omega_k) = \lambda$. Moreover, v is an eigenvector, so $v \neq 0$. Therefore, there exists at least one k such that $v_k^T v \neq 0$, which implies $\beta_\alpha(\omega_k) = \lambda$.

□

2.4 Time discretization

In the previous section, we introduced the operator Π_α that could meet our eigenvalue requirements and replace the matrix C in Krylov iteration. However, exact computation of the value $\Pi_\alpha r$ is impossible for two reasons: it demands an analytical solution $y(\cdot; r)$ to the semi-discrete wave equation (2.6) as well as the computation of an integral.

In numerical method, we replace the integral in the definition of Π_α by the rectangular rule. Since we only need the values of the function $y(\cdot; r)$ at quadrature nodes, we can discretize the problem (2.6) in time and compute an approximation of the exact solution at those points using the finite difference method.

Lemma 2.17 (Finite difference method)**.** *For a given function $y \in C^4([a,b])$ and a uniform mesh $(t_l := a + \tau l)_{l=0}^{L}$ with $L+1 \in \mathbb{N}$ nodes and mesh-size $\tau = (a-b)/L$, for all $l = 1, \dots, L-1$, we have:*

$$\ddot{y}(t_l) = \frac{y(t_{l+1}) - 2y(t_l) + y(t_{l-1})}{\tau^2} + \mathcal{O}(\tau^2).$$

Proof. This follows from a straight-forward calculation using Taylor expansion of the function y. For the details, see [INPR23, p. 93]. □

Proposition 2.18 (Finite difference method for the semi-discrete wave equation)**.** *Let $\tau > 0$ be the step size of a uniform mesh on $[0, \infty)$. By replacing the second time derivatives in (2.6) with the finite differences from Lemma 2.17, we obtain the following approximation $y_l(r)$ of $y(\tau l; r)$ for all $l \in \mathbb{N}$:*

$$\begin{aligned} y_{l+1}(r) &:= -\tau^2 M^{-1} S y_l(r) + 2y_l(r) - y_{l-1}(r) \quad \text{for } l \in \mathbb{N}, \\ y_0(r) &:= r, \quad y_1(r) := r - \frac{\tau^2}{2} M^{-1} S r \end{aligned} \tag{2.9}$$

Proof. For initial values, we have $y(0;r) = r = y_0(r)$. By Taylor expansion:

$$y(\tau; r) = y(0;r) + \tau \dot{y}(0;r) + \frac{\tau^2}{2} \ddot{y}(0;r) + \mathcal{O}(\tau^3).$$

$y(\cdot; r)$ is a solution to (2.6), thus $y(0;r) = r$, $\dot{y}(0;r) = 0$ and $\ddot{y}(0;r) = -M^{-1} S y(0;r) = -M^{-1} S r$. We get:

$$y(\tau; r) = r - \frac{\tau^2}{2} M^{-1} S r + \mathcal{O}(\tau^3) = y_1(r) + \mathcal{O}(\tau^3).$$

Replacing the second time derivative in (2.6) by finite difference leads to:

$$\begin{aligned} y(\tau(l+1); r) - 2y(\tau l; r) + y(\tau(l-1); r) &= -\tau^2 M^{-1} S y(\tau l; r) + \mathcal{O}(\tau^4), \\ y(\tau(l+1); r) &= -\tau^2 M^{-1} S y(\tau l; r) + 2y(\tau l; r) - y(\tau(l-1); r) + \mathcal{O}(\tau^4). \end{aligned}$$

Hence, $y_l(r)$ defined in (2.9) approximates $y(\tau l; r)$ for all $l \in \mathbb{N}$. □

Proposition 2.19 (Discrete operator for Krylov iteration)**.** *Let $L \in \mathbb{N}$ be the number of quadrature nodes and $T > 0$ be the upper bound of integration. Discretizing the operator Π_α (see Definition 2.15) by the rectangular rule with step size $\tau = T/L$ and approximation $y(l\tau; r) \approx y_l(r)$ leads to the discrete operator $C : \mathbb{R}^N \to \mathbb{R}^N$:*

$$r \mapsto Cr := \sum_{l=0}^{L-1} \tau \alpha(\tau l) y_l(r). \tag{2.10}$$

The discrete operator C defined in Proposition 2.19 allows us to avoid high computational costs in each Krylov step $r \mapsto Cr$. However, we have to verify the coincidence between eigenpairs of C and eigenpairs of the original problem (2.3). For this reason, we construct a discrete filter function as a discrete counterpart of β_α.

If r is an eigenvector of the matrix $M^{-1}S$ with eigenvalue ω^2, then the time-stepping (2.9) takes the following form for all $n \in \mathbb{N}$:

$$y_l(r) = q_l(\omega) r \tag{2.11}$$

with $q_l(\omega)$ being the solution to the following recursion:

$$\begin{aligned} q_{l+1}(\omega) &:= (2 - \tau^2\omega^2) q_l(\omega) - q_{l-1}(\omega) \quad \text{for } l \in \mathbb{N}, \\ q_0(\omega) &:= 1, \quad q_1(\omega) := 1 - \frac{\tau^2\omega^2}{2} \end{aligned} \tag{2.12}$$

This linear difference problem has unique solution, which will be needed later in Lemma 2.23. Therefore, we analyze the behavior of the solution as l tends to infinity.

Lemma 2.20 (Courant–Friedrichs–Lewy condition for q_l)**.** *For a given $\tau > 0$ and $\omega > 0$, the unique solution to (2.12) is:*

$$q_l(\omega) = \frac{1}{2}\left(c(\omega)^l + \overline{c(\omega)}^l \right), \tag{2.13}$$

where

$$c(\omega) := 1 - \frac{\tau^2\omega^2}{2} + \mathrm{i}\sqrt{1 - \left(1 - \frac{\tau^2\omega^2}{2}\right)^2}.$$

Therefore, if $\tau < 2/\omega$, then the CFL condition is satisfied and $|c(\omega)| = 1$, implying the boundedness of $|q_l(\omega)|$ as $l \to \infty$. Otherwise, $|q_l(\omega)|$ is unbounded.

Proof. The characteristic equation of the difference equation (2.12) is:

$$c(\omega)^2 + (\tau^2\omega^2 - 2)c(\omega) + 1 = 0.$$

If $\tau < 2/\omega$, then the solutions are:

$$c(\omega) = 1 - \frac{\tau^2\omega^2}{2} + \mathrm{i}\sqrt{1 - \left(1 - \frac{\tau^2\omega^2}{2}\right)^2} \quad \text{and} \quad \overline{c(\omega)} = 1 - \frac{\tau^2\omega^2}{2} - \mathrm{i}\sqrt{1 - \left(1 - \frac{\tau^2\omega^2}{2}\right)^2}.$$

Obviously, $|c(\omega)| = 1$. Linearity of the recursion implies that the unique solution is given by $q_l(\omega) = C_1 c(\omega)^l + C_2 \overline{c(\omega)}^l$. From the initial values, we obtain a system of two equations for C_1 and C_2 with the solution $C_1 = C_2 = 1/2$.

Therefore, representation (2.13) holds. If $\tau < 2/\omega$, then $\mathfrak{Re}(c(\omega)) < 0$, and (2.13) remains bounded as $l \to \infty$.

Otherwise, if $\tau > 2/\omega$, then the roots of the characteristic equation are real:

$$c(\omega) = 1 - \frac{\tau^2\omega^2}{2} \pm \sqrt{\left(1 - \frac{\tau^2\omega^2}{2}\right)^2 - 1}.$$

One of the roots is smaller than -1, so the solution to the linear difference equation cannot be bounded as $l \to \infty$. □

We recall the recursive definition of Chebyshev polynomials on the interval $[-1,1]$ [Nan23, p. 23] and observe connection between Chebyshev polynomials mapped to the interval $[0,4/\tau^2]$ and $q_l(\omega)$.

Definition 2.21 (Chebyshev polynomials)**.** *For all $l \in \mathbb{N}$, the Chebyshev polynomials $T_l : [-1,1] \to \mathbb{R}$ are defined as:*

$$T_0(x) := 1, \quad T_1(x) := x, \quad T_l(x) := 2xT_l(x) - T_{l-1}(x) \text{ for } l \geqslant 2.$$

Lemma 2.22. *Let $\tau > 0$. For $l \in \mathbb{N}$, let $\hat{T}_l : [0,4/\tau^2] \to \mathbb{R}$ denote the l-th Chebyshev polynomial on $[0,4/\tau^2]$, i.e., $\hat{T}_l := T_l \circ \Psi$ with affine mapping*

$$\Psi : \begin{cases} \left[0, \frac{4}{\tau^2}\right] & \to [-1,1] \\ x & \mapsto \frac{\tau^2 x}{2} - 1 \end{cases}$$

and T_l Chebyshev polynomials from Definition 2.21. Then for all $l \in \mathbb{N}$ and q_l defined in (2.12) holds:

$$q_l(\omega) = (-1)^l \hat{T}_l(\omega^2) \text{ for all } \omega \in \left[0, \frac{2}{\tau}\right].$$

Proof. From the definition of $\hat{T}_l$ follows recursion:

$$\hat{T}_0(x) \equiv 1, \quad \hat{T}_1(x) = \frac{\tau^2}{2}x - 1, \quad \hat{T}_l(x) = 2\left(\frac{\tau^2}{2}x - 1\right)\hat{T}_l(x) - \hat{T}_{l-1}(x) \text{ for } l \geqslant 2. \tag{2.14}$$

We prove the claim by induction. The case $l = 0$ is trivial. For $l = 1$, we have:

$$\hat{T}_1(\omega^2) = \frac{\tau^2\omega^2}{2} - 1 = -q_1(\omega).$$

In the induction step from (2.14) follows:

$$\begin{aligned} \hat{T}_{l+1}(\omega^2) &= \left(\tau^2\omega^2 - 2\right)\hat{T}_l(\omega^2) - T_{l-1}(\omega^2) \\ &= \left(\tau^2\omega^2 - 2\right)(-1)^l q_l(\omega) - (-1)^{l-1} q_{l-1}(\omega) \\ &= (-1)^{l+1}\left(\left(2 - \tau^2\omega^2\right)q_l(\omega) - q_{l-1}(\omega)\right) = (-1)^{l+1} q_{l+1}(\omega). \end{aligned}$$

This concludes the proof. □

For the following lemma note, that the matrix $M^{-1}S$ has a spectral decomposition $M^{-1}S = V\mathrm{diag}(\omega_1^2, \ldots, \omega_N^2)V^T$, so there exists its square root $\sqrt{M^{-1}S} = V\mathrm{diag}(\omega_1, \ldots, \omega_N)V^T$.

Lemma 2.23. *Let $L \in \mathbb{N}$ be the number of quadrature nodes, T be the upper bound of the integral, and $\tau = T/L$, as in Proposition 2.19. Let $q_l(\omega)$ be defined for all $l \in \mathbb{N}$ and $\omega \in \mathbb{R}$ by (2.12). We define the discrete filter function $\tilde{\beta}_\alpha : [0,\infty) \to \mathbb{R}$ as follows:*

$$\tilde{\beta}_\alpha(\omega) := \sum_{l=0}^{L-1} \tau\alpha(\tau l) q_l(\omega). \tag{2.15}$$

1. *$\tilde{\beta}_\alpha$ is a polynomial of degree $L-1$ in ω^2.*
2. *For C defined in Proposition 2.19, $C = \tilde{\beta}_\alpha\left(\sqrt{M^{-1}S}\right)$.*
3. *If (ω^2, v) is an eigenpair of the original problem (2.3), then $(\tilde{\beta}_\alpha(\omega), v)$ is an eigenpair of C.*
4. *If (λ, v) is an eigenpair of C, then there exists $\omega \in \mathbb{R}$ such that ω^2 is an eigenvalue of (2.3), $\tilde{\beta}_\alpha(\omega) = \lambda$, and*

$$v \in \bigoplus \{U : \exists \omega \, U \text{ is eigenspace to eigenvalue } \omega \text{ and } \tilde{\beta}_\alpha(\omega) = \lambda\}.$$

Proof. The first claim follows directly from the construction of $q_l(\omega)$ (2.12). Let $V := (v_1, \ldots, v_N)$ be an orthonormal basis of eigenvectors corresponding to eigenvalues $\omega_1^2, \ldots, \omega_N^2$ of $M^{-1}S$. The spectral decomposition of $M^{-1}S$ is $M^{-1}S = V\mathrm{diag}(\omega_1^2, \ldots, \omega_N^2)V^T$. Since (2.9) is a linear difference equation, for arbitrary $r \in \mathbb{R}^N$ and for all $l \in \mathbb{N}$ we obtain:

$$y_l(r) = y_l\left(\sum_{j=1}^{N}(r^T v_j)v_j\right) = \sum_{j=1}^{N}(r^T v_j)y_l(v_j) \overset{(2.11)}{=} \sum_{j=1}^{N}(r^T v_j)q_l(\omega_j)v_j.$$

Therefore, by the definition of C:

$$\begin{aligned} Cr &= \sum_{l=0}^{L-1}\tau\alpha(l\tau)y_l(r) = \sum_{j=1}^{N}\sum_{l=0}^{L-1}\tau\alpha(l\tau)q_l(\omega_j)(r^T v_j)v_j \\ &= \sum_{j=1}^{N}\tilde{\beta}_\alpha(\omega_j)(r^T v_j)v_j = V\mathrm{diag}\left(\tilde{\beta}_\alpha(\omega_1), \ldots, \tilde{\beta}_\alpha(\omega_N)\right)V^T r = \tilde{\beta}_\alpha\left(\sqrt{M^{-1}S}\right)r. \end{aligned}$$

So the second statement holds. The third and fourth claims follow from the second one. Let $\tilde{\beta}_\alpha(\omega) = \sum_{l=0}^{L-1} c_l\omega^{2l}$ for some $c_0, \ldots, c_{L-1} \in \mathbb{R}$. If v is an eigenvector of (2.3) to eigenvalue ω^2, then:

$$Cv = \tilde{\beta}_\alpha\left(\sqrt{M^{-1}S}\right)v = \sum_{l=0}^{L-1} c_l(M^{-1}S)^l v = \sum_{l=0}^{L-1} c_l\omega^{2l}v = \tilde{\beta}_\alpha(\omega)v,$$

so $\tilde{\beta}_\alpha(\omega)$ is an eigenvalue of C to v. If (λ, v) is an eigenpair of C, then for all $k = 0, \ldots, N$ holds:

$$\begin{aligned} 0 = (Cv - \lambda v)v_k &= \left(\left(\tilde{\beta}_\alpha\left(\sqrt{M^{-1}S}\right)v_k\right)^T v - \lambda v^T v_k\right) \\ &= \tilde{\beta}_\alpha(\omega_k)v_k^T v - \lambda v^T v_k = \left(\tilde{\beta}_\alpha(\omega_k) - \lambda\right)v^T v_k. \end{aligned}$$

Since $v^T v_k \neq 0$ for some $k = 1, \ldots, N$, it follows that $\lambda = \tilde{\beta}_\alpha(\omega_k)$ and v belongs to the sum of all eigenspaces corresponding to eigenvalues ω^2, such that $\tilde{\beta}_\alpha(\omega) = \lambda$. □

2.5 Algorithm

Now that we have constructed the operator C meeting our requirements, we can reformulate Algorithm 3 into an implementable form.

Algorithm 4 Krylov eigenvalue solver with filtered time-domain

Input: weight function α, start vector r with $\|r\|_2 = 1$, M^{-1}, S, dimension m of the Krylov space, time-step τ, number of time-steps L

$b_0 \leftarrow r$

for $k = 1, \dots, m-1$ **do** ▷ Krylov loop

  $y_0 \leftarrow b_{k-1}$, $y_1 \leftarrow b_{k-1} - \frac{\tau^2}{2} M^{-1} S b_{k-1}$

  $b_k \leftarrow \tau\alpha(0) y_0 + \tau\alpha(\tau) y_1$ ▷ First two summands in (2.10)

  for $l = 2, \dots L-1$ **do** ▷ Time-stepping in each Krylov step

    $y_l \leftarrow -\tau^2 M^{-1} S y_{l-1} + 2y_{l-1} - y_{l-2}$ ▷ y_l represents $y_l(r)$ from (2.9)

    $b_k \leftarrow b_k + \tau\alpha(\tau l) y_l$ ▷ Computation of $b_{k-1} \mapsto b_k := C b_{k-1}$

  end for

  $b_k \leftarrow b_k - \sum_{i=0}^{k-1} (b_i^T b_k) b_i$ ▷ Gram-Schmidt orthogonalization

  if $\|b_k\|_2 \neq 0$ **then**

    $b_k \leftarrow b_k / \|b_k\|_2$ ▷ Normalization

  else

    $m \leftarrow k$ ▷ $b_k \in \operatorname{span}\{b_0, \dots, b_{k-1}\}$. Exact eigenspace is a subspace of $\mathcal{K}_k(C; r)$.

    break

  end if

end for

$B_m \leftarrow (b_0, \dots, b_{m-1})$ ▷ Projection matrix

$A \leftarrow B_m^T M^{-1} S B_m$

solve $A v_m = \omega_m^2 v_m$ with a direct solver

The chosen time-step τ should fulfill the CFL condition (see Lemma 2.20) for the largest absolute value among the eigenvalues ω^2 of (2.3). The computational complexity of this method depends on the number of time-steps L (and thus by the end time of integration $T = \tau L$) and the dimension of the Krylov space m. Notice that in the presented algorithm, we can easily increase the dimension of the Krylov space without incurring significantly higher computational costs, since increasing m by $\hat{m}$ requires only $\hat{m}$ additional iterations of the Krylov loop and solving the eigenvalue problem with power iteration. For more details about the stopping criterion, we refer to [NW24, p. 8–9]. In the next chapter, we will discuss the choice of the weight function α.

3 Choice of the weight function

To fully enable the functionality of the Krylov eigenvalue solver based on the filtered time domain, we need an appropriate weight function α. The first section of this chapter follows the choice in [NW24]. In subsequent sections, we discuss other numerical methods for selecting the weight function.

The entire spectrum of eigenvalues ω^2 of the discrete approximation (2.3) of the negative Laplacian eigenvalue problem is included within some interval $\left[0, \omega_{\text{end}}^2\right]$. We search for eigenvalues within a specified region of interest $\left[\omega_{\min}^2, \omega_{\max}^2\right] \subset \left[0, \omega_{\text{end}}^2\right]$. By employing the Krylov eigenvalue solver (Algorithm 4), we can compute eigenvalues ω^2 such that the absolute value of the discrete filter function $\left|\tilde{\beta}_\alpha(\omega)\right|$ is maximized.

Consequently, our goal is to construct a weight function α such that the filter function $|\beta_\alpha|$ (2.8) or its discrete equivalent $\left|\tilde{\beta}_\alpha\right|$ (2.15) takes on larger values if $\omega \in [\omega_{\min}, \omega_{\max}]$ and is close to 0 otherwise. Of course, we can require this only in the controlled interval $[0, \omega_{\text{end}}]$, since, according to Lemma 2.23, $\tilde{\beta}_\alpha$ is a polynomial. Moreover, in Definition 2.15 we required that the function α is piecewise continuous and compactly supported.

3.1 Inverse Fourier transform method

In this section, we follow the choice in [NW24]. The first presented method of selecting a weight function is based on the observation that the mapping $\alpha \mapsto \beta_\alpha$ is somehow similar to the Fourier transform.

Lemma 3.1. *Let α be a piecewise continuous weight function with compact support and $\alpha \in L^1([0,\infty))$. Let $\hat{\alpha} : \mathbb{R} \to \mathbb{R}$ denote its symmetric extension, i.e., $\hat{\alpha}(-t) := \alpha(t)$ for all $t \in \mathbb{R}_+$. Then, for all $s \in \mathbb{R}$:*

$$\beta_\alpha(s) = \sqrt{\frac{\pi}{2}} \mathcal{F}\left[\hat{\alpha}\right](s),$$

where $\mathcal{F}$ denotes the Fourier transform and β_α denotes the filter function defined in (2.8).

Proof. Obviously, if $\alpha \in L^1([0,\infty))$, then $\hat{\alpha} \in L^1(\mathbb{R})$. From (2.8), it follows:

$$\begin{aligned}\beta_\alpha(s) &= \int_0^\infty \alpha(t)\cos(st)\,dt = \frac{1}{2}\int_{-\infty}^\infty \hat{\alpha}(t)\cos(st)\,dt + \frac{\mathrm{i}}{2}\underbrace{\int_{-\infty}^\infty \hat{\alpha}(t)\sin(-st)\,dt}_{=0} \\ &= \frac{1}{2}\int_{-\infty}^\infty \hat{\alpha}(t)e^{-\mathrm{i}st}\,dt = \sqrt{\frac{\pi}{2}}\mathcal{F}\left[\hat{\alpha}\right](s).\end{aligned}$$

The sine-integral is 0, since $\hat{\alpha}$ is an even function and $t \mapsto \sin(-st)$ is an odd function. This proves the desired identity. □

This motivates the choice of the function α as the truncated inverse Fourier transform. As the goal function to be the filter function β_α, we choose the symmetrical indicator function of the region of interest: $\chi_{[-\omega_{\max},-\omega_{\min}]\cup[\omega_{\min},\omega_{\max}]}$. Thanks to the symmetry of this function, its inverse Fourier transform is real-valued.

Theorem 3.2. *Truncation of the scaled inverse Fourier transform of the symmetric indicator function $\chi_{[-\omega_{\max},-\omega_{\min}]\cup[\omega_{\min},\omega_{\max}]}$ to a compact interval $[0,T]$, $T>0$, leads to the following weight function:*

$$\alpha(t) = \begin{cases} \frac{2(\omega_{\max}-\omega_{\min})}{\pi}, & t = 0, \\ \frac{4}{\pi t}\sin\left(\frac{t}{2}\left(\omega_{\max}-\omega_{\min}\right)\right)\cos\left(\frac{t}{2}\left(\omega_{\max}+\omega_{\min}\right)\right), & t\in(0,T], \\ 0, & \textit{else.} \end{cases} \tag{3.1}$$

Proof. Following Lemma 3.1, we compute:

$$\begin{aligned} \sqrt{\frac{2}{\pi}}\mathcal{F}^{-1}\left[\chi_{[-\omega_{\max},-\omega_{\min}]\cup[\omega_{\min},\omega_{\max}]}\right](t) &= \frac{1}{\pi}\int_{-\infty}^{\infty}\chi_{[\omega_{\min},\omega_{\max}]}e^{\mathrm{i}st}\,ds \\ &= \frac{1}{\pi}\left(\int_{-\omega_{\max}}^{-\omega_{\min}}\cos(st)\,ds+\int_{\omega_{\min}}^{\omega_{\max}}\cos(st)\,ds\right) \\ &+ \frac{\mathrm{i}}{\pi}\underbrace{\left(\int_{-\omega_{\max}}^{-\omega_{\min}}\sin(st)\,ds+\int_{\omega_{\min}}^{\omega_{\max}}\sin(st)\,ds\right)}_{=0} \\ &= \frac{2}{\pi}\int_{\omega_{\min}}^{\omega_{\max}}\cos(st)\,ds \\ &= \frac{2}{\pi t}\left(\sin\left(t\omega_{\max}\right)-\sin\left(t\omega_{\min}\right)\right) \\ &= \frac{4}{\pi t}\sin\left(\frac{t}{2}\left(\omega_{\max}-\omega_{\min}\right)\right)\cos\left(\frac{t}{2}\left(\omega_{\max}+\omega_{\min}\right)\right). \end{aligned}$$

The obtained function is real-valued, but does not have compact support. Therefore, we truncate it to the finite interval $(0,T]$ and set $\alpha|_{(T,\infty)} := 0$. Since the limit from the right as $t\to 0^+$ is finite, we can right-continuously in 0 extend the function to the interval $[0,\infty)$. This leads to representation (3.1). □

Since the discretization (2.15) of the filter function β_α truncates the time-interval, we truncate the weight function α to the same time $T>0$ as assumed in $\tilde{\beta}_\alpha$. Of course, replacing the exact improper integral in the definition of β_α (2.8) by a quadrature rule on a finite time-interval impacts the function, and therefore, the discrete filter function $\tilde{\beta}_\alpha$ cannot be exactly the target indicator function $\chi_{[\omega_{\min},\omega_{\max}]}$. However, for our purposes, it is sufficient if $\tilde{\beta}_\alpha$ takes on larger values in the region of interest $[\omega_{\min},\omega_{\max}]$ and is close to 0 otherwise.

The computational costs of the Krylov eigenvalue solver depend on the number of time-steps $L = T/\tau$. The CFL condition (see Lemma 2.20) allows us to choose time-step at most as $\tau = 2/\omega_{\text{end}}$.

We choose the setup in [NW24, section 3.1.2]: $\tau = 0.0056$, $\omega_{\text{end}} = 2/\tau \approx 357.14$, and varying end-times T and target intervals $[\omega_{\min}, \omega_{\max}]$. Results are presented in Figures 3.1 and 3.2. In each figure, the first plot presents the whole controlled interval $[0, \omega_{\text{end}}]$, and the second plot shows the behavior of the discrete filter function in the region of interest compared to the target indicator function.

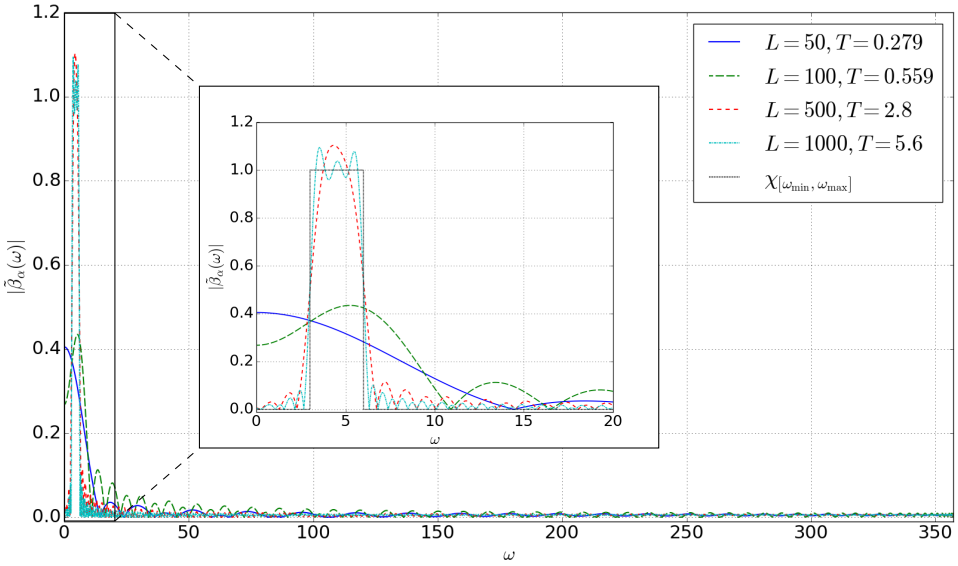


Figure 3.1: Plots of the function $\left|\tilde{\beta}_\alpha(\omega)\right|$ with the weight function α obtained by truncated inverse Fourier transform (3.1). The target interval is $[\omega_{\min}, \omega_{\max}] = [3, 6]$, the time-step is $\tau = 0.0056$, and $\omega_{\text{end}} = 2/\tau \approx 357.14$. We vary the number of time-steps L.

We can observe that for small values of L, the discrete filter function $\tilde{\beta}_\alpha$ at the end of the controlled interval (for $\omega \approx \omega_{\text{end}}$) may exceed its values in the target interval, which is unacceptable. Numerical experiments showed that increasing the number of time steps L (and thus increasing the degree of the polynomial $\tilde{\beta}_\alpha$) causes better behavior in both the target interval and the neighborhood of ω_{end} and solves this issue. However, increasing L is undesirable since it increases the computational costs of the algorithm.

It's worth noticing that the presented method can be extended for target functions other than the indicator. An arbitrary target function γ on the interval $[0, \infty)$ can be symmetrically extended to the whole real domain. This always yields a real-valued inverse Fourier transform. Truncation of this function to the interval $[0, T]$ fulfills the requirements set on the weight function.

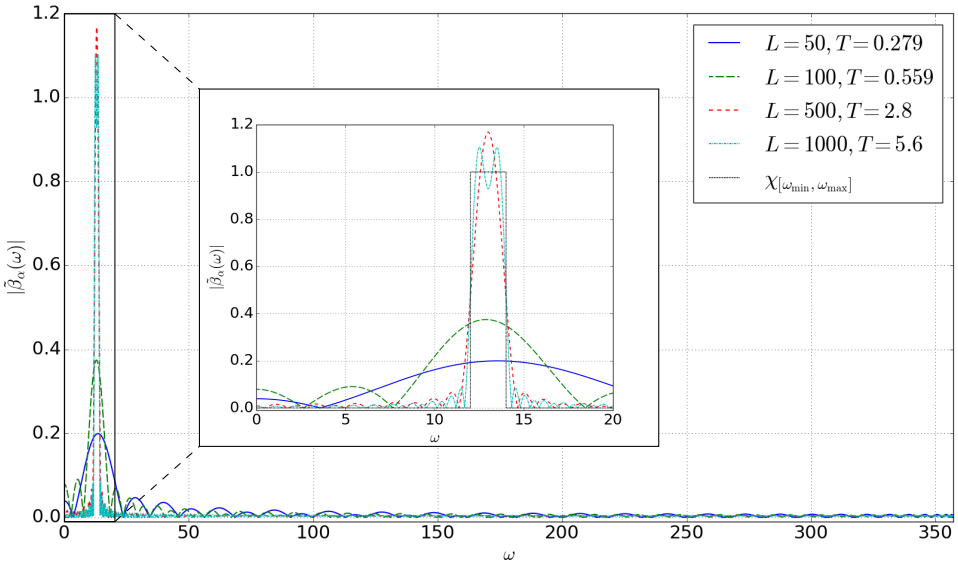


Figure 3.2: Plots of the function $\left|\tilde{\beta}_\alpha(\omega)\right|$ with the weight function α obtained by truncated inverse Fourier transform (3.1). The target interval is $[\omega_{\min}, \omega_{\max}] = [12, 14]$, the time-step is $\tau = 0.0056$, and $\omega_{\text{end}} = 2/\tau \approx 357.14$. We vary the number of time-steps L.

3.2 Collocation method

The definitions of the discrete operator C for Krylov iteration (2.10) and of the discrete filter function $\tilde{\beta}_\alpha$ (2.15) for a fixed time-step τ and number of time-steps L do not require the values of the weight function in the whole domain, but only at τl for $l = 0, \ldots, L-1$. Thus, we can reduce the problem to finding a vector $\vec{\alpha} \in \mathbb{R}^L$, such that $\vec{\alpha}_l$ represents the value $\alpha(\tau l)$ for all $l = 0, \ldots, L-1$.

Definition 3.3. *Let $\tau > 0$ be the time-step, $L \in \mathbb{N}$ be the number of time-steps and let $q_l(\omega)$ be defined for all $l \in \mathbb{N}$ and $\omega \in \mathbb{R}$ by (2.12). Let $\vec{\alpha} \in \mathbb{R}^L$ be a weight vector. Then, the discrete filter function $\tilde{\beta}_{\vec{\alpha}} : [0, \infty) \to \mathbb{R}$ is defined as:*

$$\tilde{\beta}_{\vec{\alpha}}(\omega) := \tau \left(q_0(\omega), \ldots, q_{L-1}(\omega)\right) \vec{\alpha}. \tag{3.2}$$

Note that for a given weight function $\alpha : [0, \omega_{\text{end}}] \to \mathbb{R}$ and evaluation vector $\vec{\alpha} := (\alpha(\tau l))_{l=0}^{L-1}$, the definition of $\tilde{\beta}_\alpha$ in Lemma 2.23 is equivalent to Definition 3.3, i.e., $\tilde{\beta}_\alpha = \tilde{\beta}_{\vec{\alpha}}$.

We fix an arbitrary target function γ that fulfills our requirements for the discrete filter function in the controlled interval. We select L nodes in the controlled interval, in the simplest setting equidistant, and formulate our problem as a polynomial collocation problem with $\{q_l : l = 0, \ldots, L-1\}$ basis of the space of polynomials up to degree $L-1$ in ω^2.

Definition 3.4. *For a given number of time-steps $L \in \mathbb{N}$ and $K \in \mathbb{N}$ collocation nodes $0 \leqslant \omega_0 < \cdots < \omega_{K-1} \leqslant \omega_{\text{end}}$ we define the evaluation matrix $Q \in \mathbb{R}^{K \times L}$ as $Q_{kl} := q_l(\omega_k)$ for all $k = 0, \ldots, K-1$ and $l = 0, \ldots, L-1$.*

Selecting L equidistant collocation nodes $\omega_l := l\omega_{\text{end}}/(L-1)$, $l = 0, \ldots, L-1$, in the controlled interval $[0, \omega_{\text{end}}]$ leads to a collocation problem to find $\vec{\alpha} \in \mathbb{R}^L$, such that:

$$\forall k = 0, \ldots, L-1: \quad \tilde{\beta}_{\vec{\alpha}}(\omega_k) = \gamma(\omega_k).$$

This problem is equivalent to the linear system of equations:

$$Q\vec{\alpha} = \frac{1}{\tau}\left(\gamma(\omega_k)\right)_{k=0}^{L-1} \tag{3.3}$$

with matrix Q from Definition 3.4.

As an example, we choose the same parameters as in the previous section, especially the target function as an indicator of the target interval $\gamma = \chi_{[\omega_{\min}, \omega_{\max}]}$. Plots of the obtained discrete filter functions for two values of $L = 10$ and $L = 25$ are shown in Figure 3.3. We solve the system of linear equations (3.3) with the `numpy.linalg.solve()` function in Python.

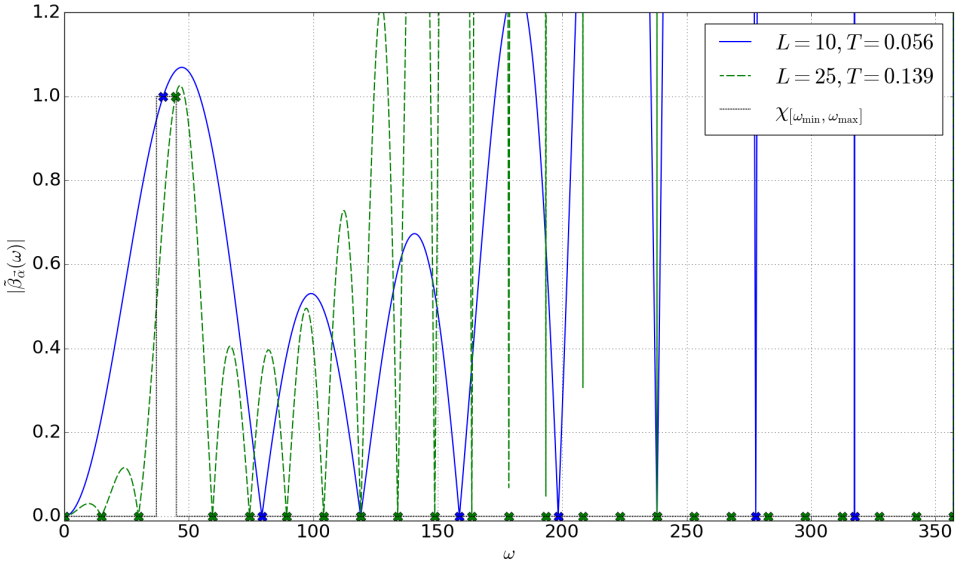


Figure 3.3: Plots of the function $\left|\tilde{\beta}_{\vec{\alpha}}\right|$ with $\vec{\alpha}$ obtained by collocation (3.3). The target interval is $[\omega_{\min}, \omega_{\max}] = [39, 45]$, the time-step is $\tau = 0.0056$, and $\omega_{\text{end}} = 2/\tau \approx 357.14$. We vary the number of time-steps and collocation nodes L. Crosses represent the values of the target function at the knots.

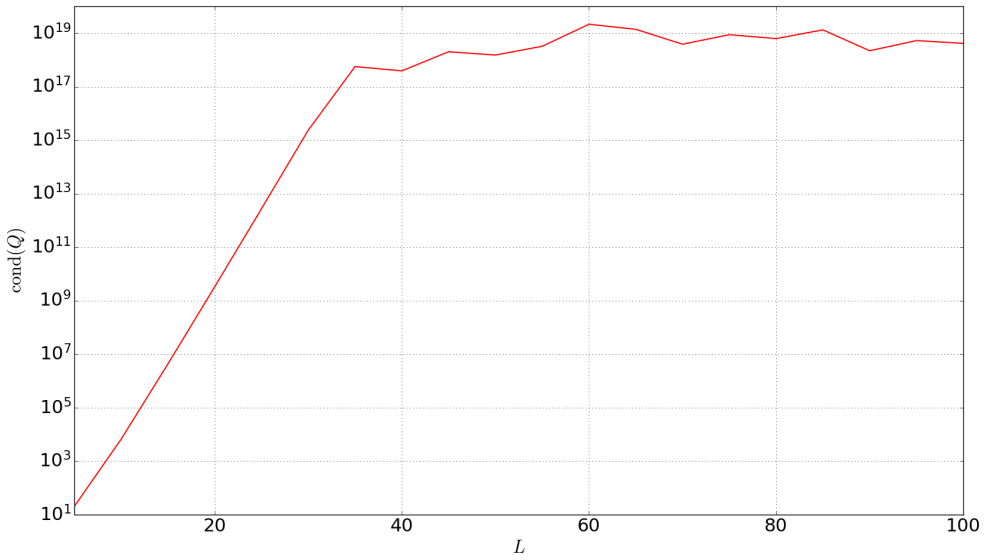


Figure 3.4: The condition number of the evaluation matrix Q (matrix norm induced by the Euclidean norm) depending on the number of collocation nodes L.

By a small number of collocation knots, intervals between nodes are large, and we can observe strong oscillation of the polynomial, especially for larger values of ω. Increasing the number of nodes does not improve the behavior for two reasons. Firstly, this increases the degree of the polynomial, which compounds the problem despite shorter intervals between nodes. Secondly, higher values of L rapidly worsen the conditioning of the matrix Q. Figure 3.4 presents $\mathrm{cond}(Q) := \|Q\|_2\|Q^{-1}\|_2$ (where $\|\cdot\|_2$ denotes the induced matrix norm by the Euclidean norm) depending on the problem's size L.

Because equidistant nodes in the collocation method do not produce desired results, we attempt to replace them. From Lemma 2.22, we know that functions $q_l(\omega)$ are polynomials in ω^2. The next proposition shows that the natural choice is the Chebyshev collocation.

Proposition 3.5 (Chebyshev nodes)**.** *For $L \in \mathbb{N}$, Chebyshev nodes on the interval $(-1, 1)$ are given by:*

$$t_l := \cos\left(\frac{2l+1}{2L}\pi\right) \text{ for } l = 0, \ldots, L-1.$$

1. *The Chebyshev polynomial T_L (see Definition 2.21) has L roots in the interval $(-1, 1)$, which are exactly the Chebyshev nodes.*

2. *For $f \in C^L([a, b])$, arbitrary collocation nodes $x_0, \ldots, x_{L-1} \in [a, b]$ and Lagrange collocation polynomial $p \in \mathbb{P}_{L-1}$ such that $p(x_l) = f(x_l)$ for all $l = 0, \ldots, L-1$, there holds the error estimation:*

$$\|f - p\|_\infty \leqslant \frac{\|f^{(L+1)}\|_\infty}{(L+1)!} \max_{x\in[a,b]} \prod_{l=0}^{L-1} |x - x_l| \,. \tag{3.4}$$

3. *The affine transformation* $\Psi : [-1,1] \to [a,b]$, $\Psi(x) := (a+b+x(b-a))/2$ *maps Chebyshev nodes* $t_0, \dots, t_{L-1}$ *onto* $\Psi(t_0), \dots, \Psi(t_{L-1}) \in [a,b]$. *These nodes minimize the product in the error estimation, i.e.,*

$$\min_{(x_0,\dots,x_{L-1})\in[a,b]^L} \max_{x\in[a,b]} \prod_{l=0}^{L-1} |x - x_l| = \max_{x\in[a,b]} \prod_{l=0}^{L-1} |x - \Psi(t_l)| = \left(\frac{b-a}{2}\right)^L 2^{1-L}.$$

Proof. We refer to [Nan23, p. 23–24]. □

For a fixed time-step $\tau > 0$, we choose collocation nodes $\omega_0, \dots, \omega_{L-1} \in [0, 2/\tau]$, such that $\omega_0^2, \dots, \omega_{L-1}^2$ are Chebyshev nodes in the interval $\left[0, 4/\tau^2\right]$, and solve the linear system of equations (3.3). Figures 3.5 and 3.6 show the obtained discrete filter functions $\left|\tilde{\beta}_{\vec{\alpha}}\right|$ for different numbers of time-steps L and target intervals $[\omega_{\min}, \omega_{\max}]$. We compare the behavior of the obtained discrete filter function with the target function and the inverse Fourier transformation (3.1) method.

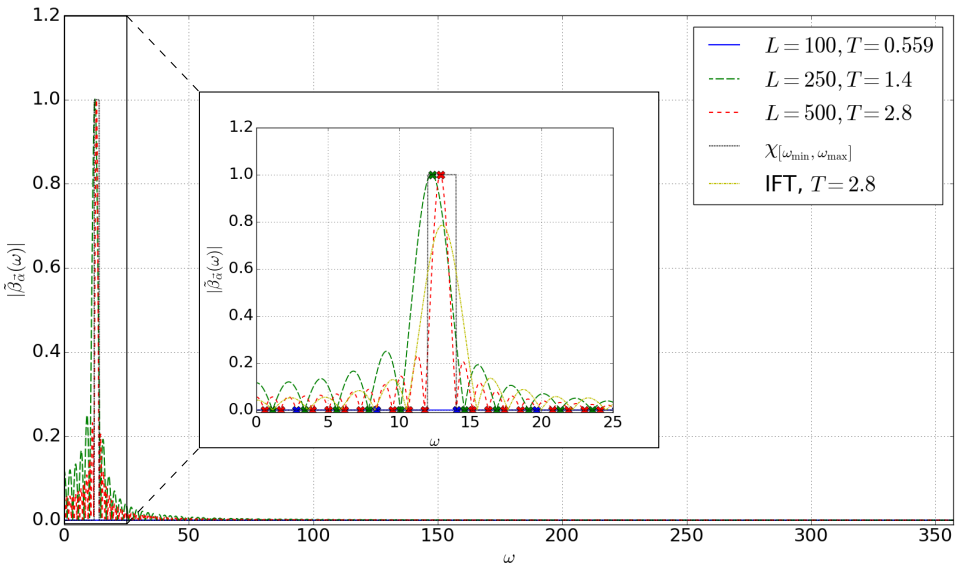


Figure 3.5: Plots of the function $\left|\tilde{\beta}_{\vec{\alpha}}\right|$ with $\vec{\alpha}$ obtained by collocation with Chebyshev nodes in ω^2. The target interval is $[\omega_{\min}, \omega_{\max}] = [12, 14]$, with the time-step $\tau = 0.0056$ and $\omega_{\text{end}} = 2/\tau \approx 357.14$. We vary the number of time-steps and collocation nodes L. Crosses represent values of the target function at the knots. The yellow curve represents the discrete filter function obtained by the inverse Fourier transform of the indicator truncated to end-time $T = 2.8$.

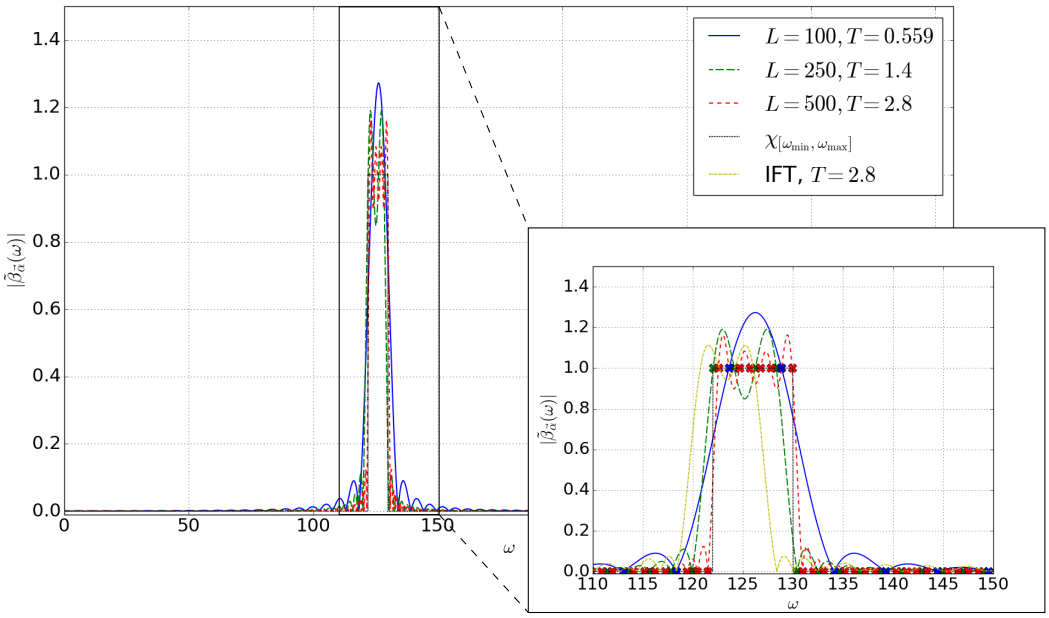


Figure 3.6: Plots of the function $\left|\tilde{\beta}_{\vec{\alpha}}\right|$ with $\vec{\alpha}$ obtained by collocation with Chebyshev nodes in ω^2. The target interval is $[\omega_{\min}, \omega_{\max}] = [122, 130]$, with the time-step $\tau = 0.0056$ and $\omega_{\text{end}} = 2/\tau \approx 357.14$. We vary the number of time-steps and collocation nodes L. Crosses represent values of the target function at the knots. The yellow curve represents the discrete filter function obtained by the inverse Fourier transform of the indicator truncated to end-time $T = 2.8$.

Error estimation for Lagrange interpolation (3.4), which was our motivation to replace equidistant nodes with Chebyshev ones, demands a function γ that is L times continuously differentiable, which is not the case for the indicator function. Therefore, in the third example, we change our target function to a C^∞ function, the Gaussian peak $g(\omega) = \exp\left(-(\omega - \omega_{\text{mid}})^2\right)$ with ω_{mid} in the middle of the region of interest. Results are presented in Figure 3.7. Again, we compare the obtained discrete filter function to one produced with the truncated inverse Fourier transform in a similar way to the derivation of (3.1).

A remarkable advantage of collocation in Chebyshev nodes is the good conditioning of the collocation matrix Q. The condition number for Chebyshev nodes is, in contrast to equidistant nodes, independent of the size of the problem L. Furthermore, the obtained discrete filter function does not oscillate strongly between collocation knots, even for an uncontinuous target function like the indicator.

Following [NW24], we assume that the sought eigenvalues of the negative Laplace operator are closer to 0 than to the end of the controlled interval ω_{end}. Chebyshev nodes in ω^2 have the highest density in the upper area of the interval $[0, \omega_{\text{end}}]$. Thus, especially for lower numbers of collocation knots and a small target interval, the entire interval $[\omega_{\min}, \omega_{\max}]$ may be contained between two adjacent nodes. This sets the right-hand

side of the collocation problem (3.3) to 0, and as a result, we get $\vec{\alpha} = 0$ and $\tilde{\beta}_{\vec{\alpha}} \equiv 0$. We can observe this problem in Figure 3.5 for $L = 100$. In this particular example, $8.4142 \approx \omega_1 < \omega_{\min} = 12 < 14 = \omega_{\max} < \omega_2 \approx 14.0213$. Furthermore, a small shift of the target interval does not impact the discrete filter function at all if the change in $\omega_{\min}$ and $\omega_{\max}$ does not exceed the distance to the nearest nodes. A similar problem can be observed for the C^∞ target function for $L = 100$ in Figure 3.7.

Obviously, with increasing L, the distances between nodes decrease, which solves this issue. However, this is undesired, since we want to choose L as small as possible due to the complexity of the algorithm. Another possibility would be to increase the number of collocation nodes K by the same number of basis functions L. This would lead to a $K \times L$ evaluation matrix with $K > L$ and a least squares problem instead of a linear system of equations.

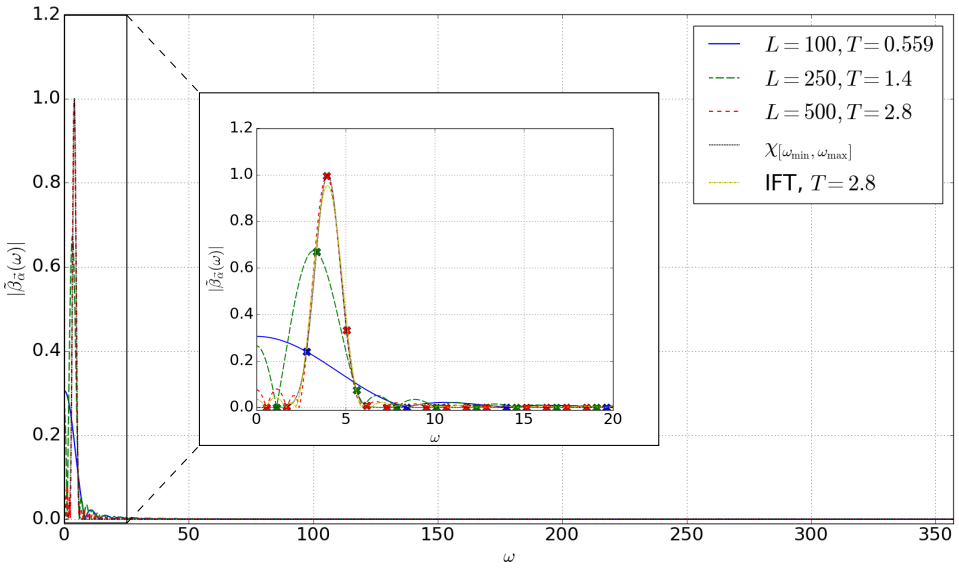


Figure 3.7: Plots of the function $\left|\tilde{\beta}_{\vec{\alpha}}\right|$ with $\vec{\alpha}$ obtained by collocation in Chebyshev nodes in ω^2. The target function is $g(\omega) = \exp\left(-(\omega - 4)^2\right)$, with a time-step of $\tau = 0.0056$ and $\omega_{\text{end}} = 2/\tau \approx 357.14$. We vary the number of time-steps and collocation nodes L. Crosses represent values of the target function at the knots. The yellow curve represents the discrete filter function obtained by the inverse Fourier transform of the indicator truncated to end-time $T = 2.8$.

3.3 Least squares method and minimization of L^2 norm

We start this section with the idea of fixing the number L of time-steps (and thus the polynomial degree) to avoid increasing the computational costs of the algorithm, while simultaneously increasing the number of collocation knots. Obviously, the obtained system

of equations is (in general) not solvable, so we use linear least squares fitting. Then, we switch to another approach – we try to find $\vec{\alpha}$ that minimizes the L^2 norm of the difference between the corresponding discrete filter function and the target function γ. Finally, Theorem 3.13 will show the connection between these two methods.

Definition 3.6 (Least squares method)**.** *Let $\tau > 0$ denote the time-step and $L \in \mathbb{N}$ denote the number of time-steps. For given nodes $\omega_0, \ldots, \omega_{K-1} \in [0, \omega_{\text{end}}]$ with $K > L$, let $Q \in \mathbb{R}^{K \times L}$ denote the evaluation matrix from Definition 3.4. For a piecewise continuous target function $\gamma : [0, \omega_{\text{end}}] \to \mathbb{R}$, we search for $\vec{\alpha} \in \mathbb{R}^L$, such that:*

$$\|Q\vec{\alpha} - c\|_2 = \min_{x \in \mathbb{R}^L} \|Qx - c\|_2, \tag{3.5}$$

where $c := 1/\tau \left(\gamma(\omega_0), \ldots, \gamma(\omega_{K-1})\right)^T$.

Note that the problem stated in Definition 3.6 is just a modification of the collocation problem (3.3) for the case $K > L$.

Proposition 3.7 (Normal equation)**.** *The problem stated in Definition 3.6 is well-defined, i.e., it has at least one solution. A vector $\vec{\alpha} \in \mathbb{R}^L$ solves (3.5) if and only if it solves the normal equation:*

$$Q^T Q\vec{\alpha} = Q^T c. \tag{3.6}$$

If the rank of the matrix Q is full, this solution is unique.

Proof. We refer to [Pra22, p. 103]. □

Another approach would be to define, for a fixed number of time-steps L and a time-step $\tau > 0$, an operator $\Phi : \mathbb{R}^L \to \mathbb{R}$ as an L^2 norm on the controlled interval of the difference between the goal function γ and the corresponding discrete filter function associated with $\vec{\alpha} \in \mathbb{R}^L$. In our method, we will attempt to determine the global minimum of this continuous operator.

Definition 3.8 (L^2 scalar product and norm)**.** *For a given controlled interval $[0, \omega_{\text{end}}]$, the L^2 scalar product is defined as: $\langle \cdot, \cdot \rangle : L^2\left([0, \omega_{\text{end}}]\right) \times L^2\left([0, \omega_{\text{end}}]\right) \to \mathbb{R}$,*

$$\langle f, g \rangle := \int_0^{\omega_{\text{end}}} f(x)g(x)\, dx.$$

The induced L^2 norm is $\|\cdot\|_{L^2} : L^2\left([0, \omega_{\text{end}}]\right) \to \mathbb{R}_{\geqslant 0}$, $\|f\|_{L^2} := \sqrt{\langle f, f \rangle}$.

Definition 3.9 (Operator Φ)**.** *Let $[0, \omega_{\text{end}}]$ be the given controlled interval, $\tau > 0$ be the time-step and L be the number of time-steps. For a given piecewise continuous (and thus L^2) goal function $\gamma : [0, \omega_{\text{end}}] \to \mathbb{R}$, we define the operator $\Phi : \mathbb{R}^L \to \mathbb{R}$ as:*

$$\Phi(v) := \left\| \tilde{\beta}_v - \gamma \right\|_{L^2}^2 = \int_0^{\omega_{\text{end}}} \left(\tilde{\beta}_v(x) - \gamma(x) \right)^2 dx$$

with $\tilde{\beta}_v$ the discrete filter function defined in (3.2).

Since the defined operator Φ mimics the squared L^2 distance between the target function and the obtained filter function, our goal is to minimize its value. The following theorem proves that it is possible without incurring high computational costs.

Theorem 3.10. *The operator Φ from Definition 3.9 is continuously differentiable. It has exactly one global minimum $\vec{\alpha} \in \mathbb{R}^L$, which is the solution to the linear system of equations:*

$$X\vec{\alpha} = d, \tag{3.7}$$

where the matrix $X \in \mathbb{R}^{L\times L}$ is defined as $X_{ij} := \langle q_i, q_j\rangle$ for all $i, j = 0, \dots, L-1$ and the vector $d \in \mathbb{R}^L$ is defined as $d_j := \langle q_j, \gamma\rangle/\tau$ for all $j = 0, \dots, L-1$.

Proof. The regularity of the function Φ follows from its definition. From (3.2), we obtain the representation:

$$\tilde{\beta}_v(x) = \tau \sum_{l=0}^{L-1} q_l(x) v_l.$$

We compute the partial derivatives of Φ:

$$\begin{aligned}
\frac{\partial\Phi}{\partial v_i}(v) &= \frac{\partial}{\partial v_i} \int_0^{\omega_{\text{end}}} \left(\tau \sum_{l=0}^{L-1} q_l(x) v_l - \gamma(x)\right)^2 dx \\
&= 2 \int_0^{\omega_{\text{end}}} \left(\tau \sum_{l=0}^{L-1} q_l(x) v_l - \gamma(x)\right) \tau q_i(x)\, dx \\
&= 2\tau \left(\left(\sum_{l=0}^{L-1} \tau v_l \int_0^{\omega_{\text{end}}} q_l(x) q_i(x)\, dx\right) - \int_0^{\omega_{\text{end}}} q_i(x)\gamma(x)\, dx\right) \\
&= 2\tau \left(\left(\tau \sum_{l=0}^{L-1} v_l \langle q_l, q_i\rangle\right) - \langle q_i, \gamma\rangle\right).
\end{aligned}$$

Therefore, the gradient $\nabla\Phi$ vanishes in $\vec{\alpha}$ if and only if:

$$\forall i = 0, \dots, L-1: \quad \tau \sum_{l=0}^{L-1} \langle q_l, q_i\rangle \vec{\alpha}_l = \langle q_i, \gamma\rangle,$$

which is equivalent to (3.7).

In the next step, we compute the second partial derivatives:

$$\frac{\partial^2\Phi}{\partial v_i \partial v_j}(v) = 2\tau \frac{\partial}{\partial v_j} \left(\left(\tau \sum_{l=0}^{L-1} v_l \langle q_l, q_i\rangle\right) - \langle q_i, \gamma\rangle\right) = 2\tau^2 \langle q_i, q_j\rangle.$$

Thus, the Hessian matrix is:

$$\nabla^2\Phi(v) = 2\tau^2 X.$$

Let us consider $B := \left(q_0|_{[0,\omega_{\text{end}}]}, \dots, q_{L-1}|_{[0,\omega_{\text{end}}]}\right)$ and $U := \operatorname{span}(B) \subset L^2([0, \omega_{\text{end}}])$. U is an L-dimensional Hilbert space with the restricted L^2 scalar product $\langle\cdot,\cdot\rangle_U = \langle\cdot,\cdot\rangle|_{U^2}$. The representation matrix of this scalar product with respect to the basis B is X. Therefore X is positive definite and especially $\nabla^2\Phi(\vec{\alpha})$ is also positive definite. This concludes, that $\vec{\alpha}$ is the unique minimum of Φ. □

We still lack a practical evaluation of the matrix X and the vector d from (3.7) to make the L^2 minimization method fully functional. Analytical computation of the integrals is out of reach here, so we replace them with the midpoint rule using K subintervals:

$$\langle f, g\rangle = \int_0^{\omega_{\text{end}}} f(\omega)g(\omega)\,d\omega \approx h\sum_{k=0}^{K-1} f(\omega_k)g(\omega_k),$$

where $h := \omega_{\text{end}}/K$ is the quadrature step-size and $\omega_k := (k+1/2)h$ for $k = 0, \dots, K-1$ are quadrature nodes. With X_h and d_h we denote the approximations of the matrix X and the vector d.

Definition 3.11. *Let K be a given number of quadrature nodes on the interval $[0, \omega_{\text{end}}]$, $h := \omega_{\text{end}}/K$ and $\omega_k := (k+1/2)h$ for all $k = 0, \dots, K-1$. For a given number of time-steps L and goal function γ, we define:*

$$X_h := \left(h\sum_{k=0}^{K-1} q_i(\omega_k)q_j(\omega_k)\right)_{i,j=0}^{L-1} \in \mathbb{R}^{L\times L} \text{ and } d_h := \left(\frac{h}{\tau}\sum_{k=0}^{K-1} q_i(\omega_k)\gamma(\omega_k)\right)_{i=0}^{L-1} \in \mathbb{R}^L.$$

Lemma 3.12. *For a given h and K, let $Q \in \mathbb{R}^{K\times L}$ be the evaluation matrix on quadrature nodes $\omega_k = (k+1/2)h$ for $k = 0, \dots, K-1$ (see Definition 3.4). Let $c := 1/\tau\,(\gamma(\omega_0), \dots, \gamma(\omega_{K-1}))^T$ as in Definition 3.6. Then, for X_h and d_h from Definition 3.11, it holds:*

$$hQ^TQ = X_h \quad \text{and} \quad hQ^Tc = d_h.$$

Proof. To show the first identity, let us consider arbitrary $i, j = 0, \dots, L-1$. From the definitions of Q and X_h follows:

$$\left(hQ^TQ\right)_{i,j} = h\sum_{k=0}^{K-1} Q_{k,i}Q_{k,j} = h\sum_{k=0}^{K-1} q_i(\omega_k)q_j(\omega_k) = (X_h)_{i,j}.$$

Similarly, we show the second claim. For $i = 0, \dots, L-1$ holds:

$$\left(hQ^Tc\right)_i = h\sum_{k=0}^{K-1} Q_{k,i}c_k = h\sum_{k=0}^{K-1} q_i(\omega_k)\frac{\gamma(\omega_k)}{\tau} = (d_h)_i\,,$$

which concludes the proof. □

We will implement the L^2 minimization using the introduced approximation of the integrals by the midpoint quadrature rule. The claim of the last lemma simplifies the preparation of the matrix X_h and the right-hand side of the equation d_h in practice. Moreover, a remarkable consequence of this lemma is the following link between obtaining $\vec{\alpha}$ by the least squares method and the normal equation (3.6) with equidistant knots, and by L^2 minimization (3.7).

Theorem 3.13 (Least squares method with equidistant knots converges to L^2 minimization)**.** *Let $\tau > 0$ be the time-step, L the number of time-steps, and $[0, \omega_{\text{end}}]$ the controlled interval. Let γ be a piecewise continuous target function. For all $h > 0$, let $K := \lfloor \omega_{\text{end}}/h \rfloor$ denote the number of equidistant knots: $\Delta_h := \omega_k := (k + 1/2)h : \quad k = 0, \dots, K-1$. With $\vec{\alpha}_h$, we denote the solution to the normal equation (see Proposition 3.7) with knots Δ_h. Let $\vec{\alpha}$ denote the solution to the L^2 minimization (3.7). Then $\lim_{h\to 0} \vec{\alpha}_h = \vec{\alpha}$.*

Proof. For every $h > 0$ and $K > L$, let Q_h be the evaluation matrix (see Definition 3.4) in knots Δ_h. According to Proposition 3.7, $\vec{\alpha}_h$ solves the normal equation:

$$Q_h^T Q_h \vec{\alpha}_h = Q_h^T c,$$

with $c := 1/\tau \left(\gamma(\omega_0), \dots, \gamma(\omega_{K-1})\right)^T$, or, equivalently:

$$hQ_h^T Q_h \vec{\alpha}_h = hQ_h^T c.$$

Thus from Lemma 3.12 follows, that $\vec{\alpha}_h$ solves:

$$X_h \vec{\alpha}_h = d_h. \tag{3.8}$$

Since γ is a piecewise continuous function on a compact interval and q_l $(l = 0, \dots, L-1)$ are polynomials, midpoint quadratures converge to the exact integrals, as h tends to 0. Therefore, $\lim_{h\to 0} X_h = X$ and $\lim_{h\to 0} d_h = d$. From (3.8) and the continuity of the matrix inversion, we conclude $\lim_{h\to 0} \vec{\alpha}_h = \vec{\alpha}$. □

As a first example, we use the least squares method with Chebyshev nodes. We recall Figure 3.5, where $L = 100$ time-steps and the same number of collocation knots produced a constant zero discrete filter function, because the whole target interval $[\omega_{\min}, \omega_{\max}]$ was contained between two collocation nodes. We use the same setting: $L = 100$, $\tau = 0.0056$, $\omega_{\text{end}} = 2/\tau$ and $[\omega_{\min}, \omega_{\max}] = [12, 14]$, but the linear least squares method and varying number of knots instead of collocation. Results are presented in Figure 3.9 compared with the goal function and the inverse Fourier method.

Introducing a larger number of nodes K with a fixed number of time-steps L yields better accuracy of the obtained discrete filter function to the goal function and does not affect the performance of the eigenvalue solver. Furthermore, thanks to the use of Chebyshev nodes, the condition number of the matrix in the normal equation $\text{cond}(Q^T Q)$ remains constant despite increasing number of nodes or time-steps, see Figure 3.8.

In the next two examples, we use L^2 minimization with the midpoint quadrature rule, equidistant nodes $\omega_k = (k+1/2)h$ for $k = 0, \dots, K-1$, $K := \lfloor \omega_{\text{end}}/h \rfloor$. Figure 3.10 presents results for the target interval $\left[\omega_{\min}^2, \omega_{\max}^2\right] = [12, 14]$, and Figure 3.11 presents results for the target interval $\left[\omega_{\min}^2, \omega_{\max}^2\right] = [122, 130]$.

In both cases, the behavior of the discrete filter function in the neighborhood of the target interval is satisfying. However, with a larger size of the problem (number of time-steps), we can observe undesired oscillation of the function $\tilde{\beta}_{\vec{\alpha}}$ at the end of the controlled interval. This is because of an increase in the condition number of the matrix $X_h = hQ^T Q$ as L gets larger. Note that $\text{cond}(X_h) = \text{cond}(Q^T Q)$. Figure 3.8 shows that by equidistant nodes (which are quadrature knots in the case of L^2 minimization and midpoint quadrature), the

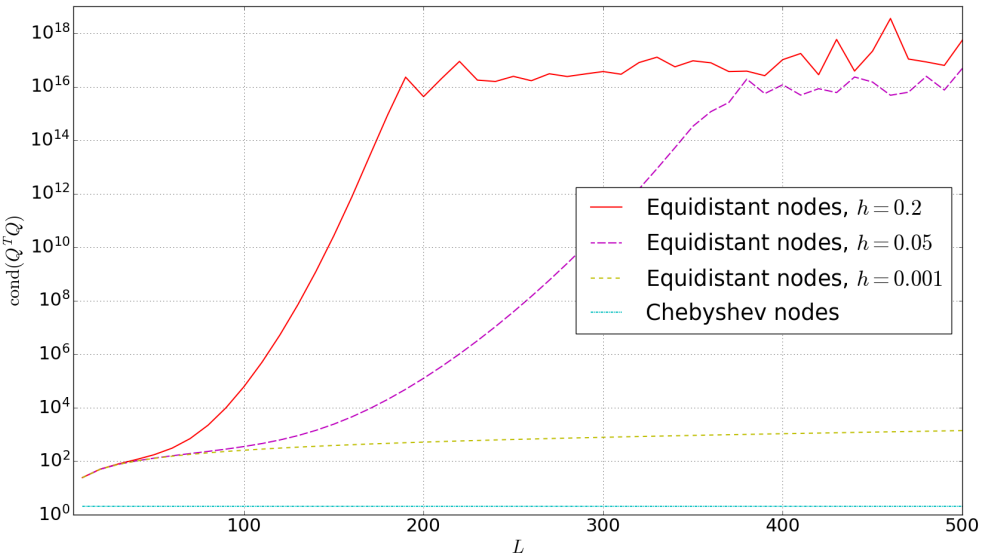


Figure 3.8: Condition number of the matrix Q^TQ (matrix norm induced by the Euclidean norm) depending on the size of the matrix L and different numbers and distributions of nodes. The number of Chebyshev nodes is equal to the number of equidistant nodes for $h = 0.2$.

problem becomes poorly conditioned if its size is too large. However, higher exactness of the quadrature slows down this effect. In Figure 3.11, we can observe that the values of the discrete filter function (red plot, with $L = 200$ and $h = 0.05$) explode as $\omega \lessapprox \omega_{\mathrm{end}}$. The use of quadrature with $h = 0.001$ (light blue plot) solves this issue despite the same value of L.

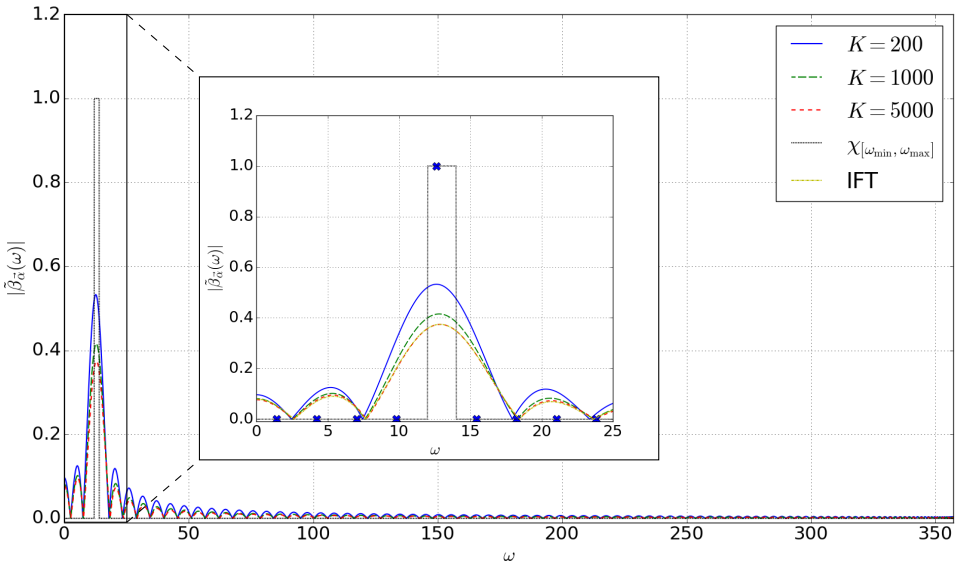


Figure 3.9: Plots of the function $\left|\tilde{\beta}_{\vec{\alpha}}\right|$ with $\vec{\alpha}$ obtained by the least squares method with Chebyshev nodes in ω^2. The target interval $[\omega_{\min}, \omega_{\max}] = [12, 14]$, time-step $\tau = 0.0056$, $\omega_{\text{end}} = 2/\tau \approx 357.14$ and fixed number of time-steps $L = 100$. We vary the number of nodes K. Crosses represent values of the target function in Chebyshev nodes for $K = 200$. The yellow curve is the discrete filter function obtained by the inverse Fourier transform of the indicator truncated to end-time $T = L\tau = 0.56$.

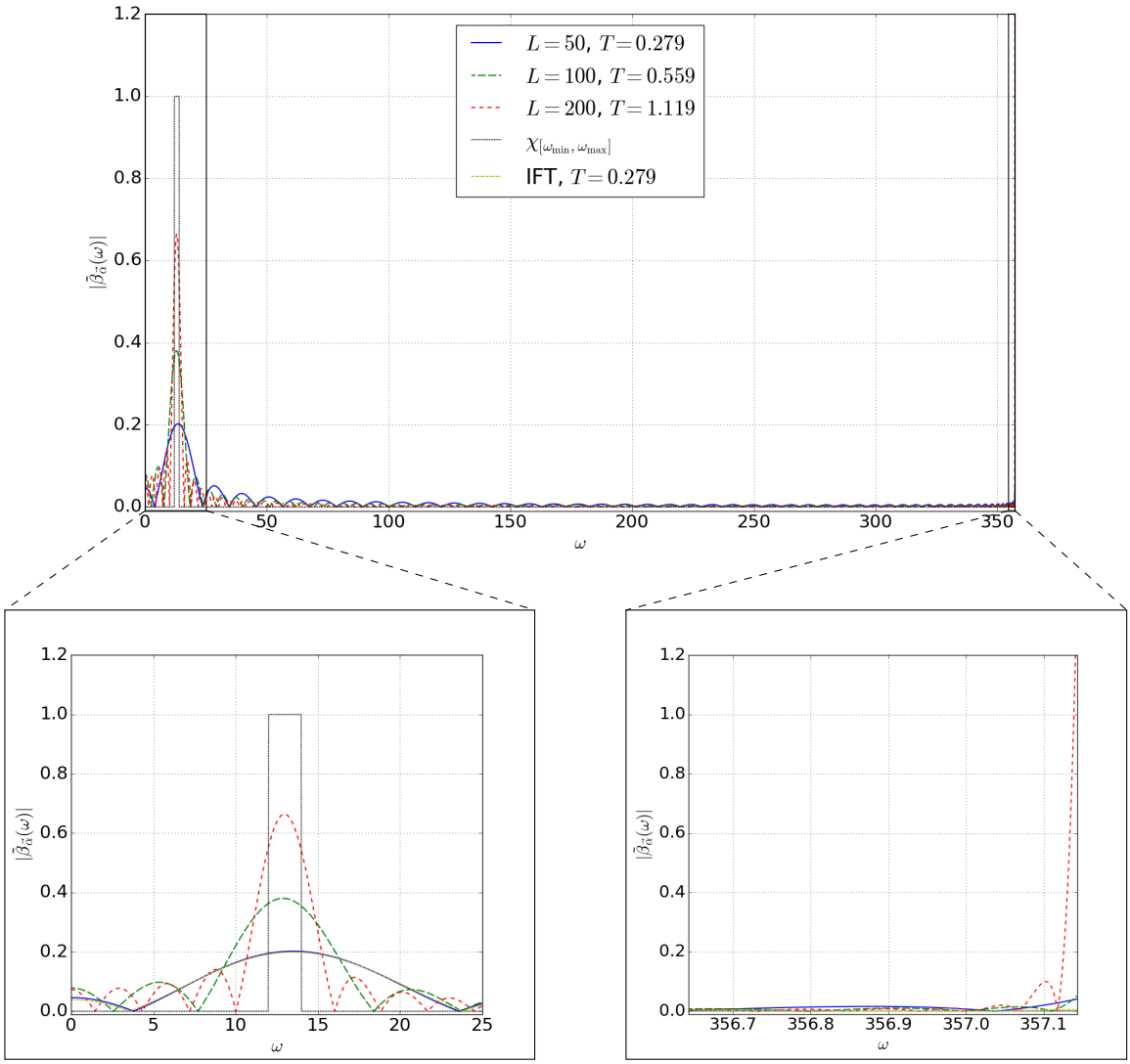


Figure 3.10: Plots of the function $\left|\tilde{\beta}_{\vec{\alpha}}\right|$ with the vector $\vec{\alpha}$ obtained by the L^2 minimization method with midpoint quadrature, $h = 0.05$. The target interval is $[\omega_{\min}, \omega_{\max}] = [12, 14]$, the time-step is $\tau = 0.0056$, and $\omega_{\text{end}} = 2/\tau \approx 357.14$. We vary the number of time-steps L. The yellow curve represents the discrete filter function obtained by the inverse Fourier transform of the indicator with end-time $T = 0.279$.

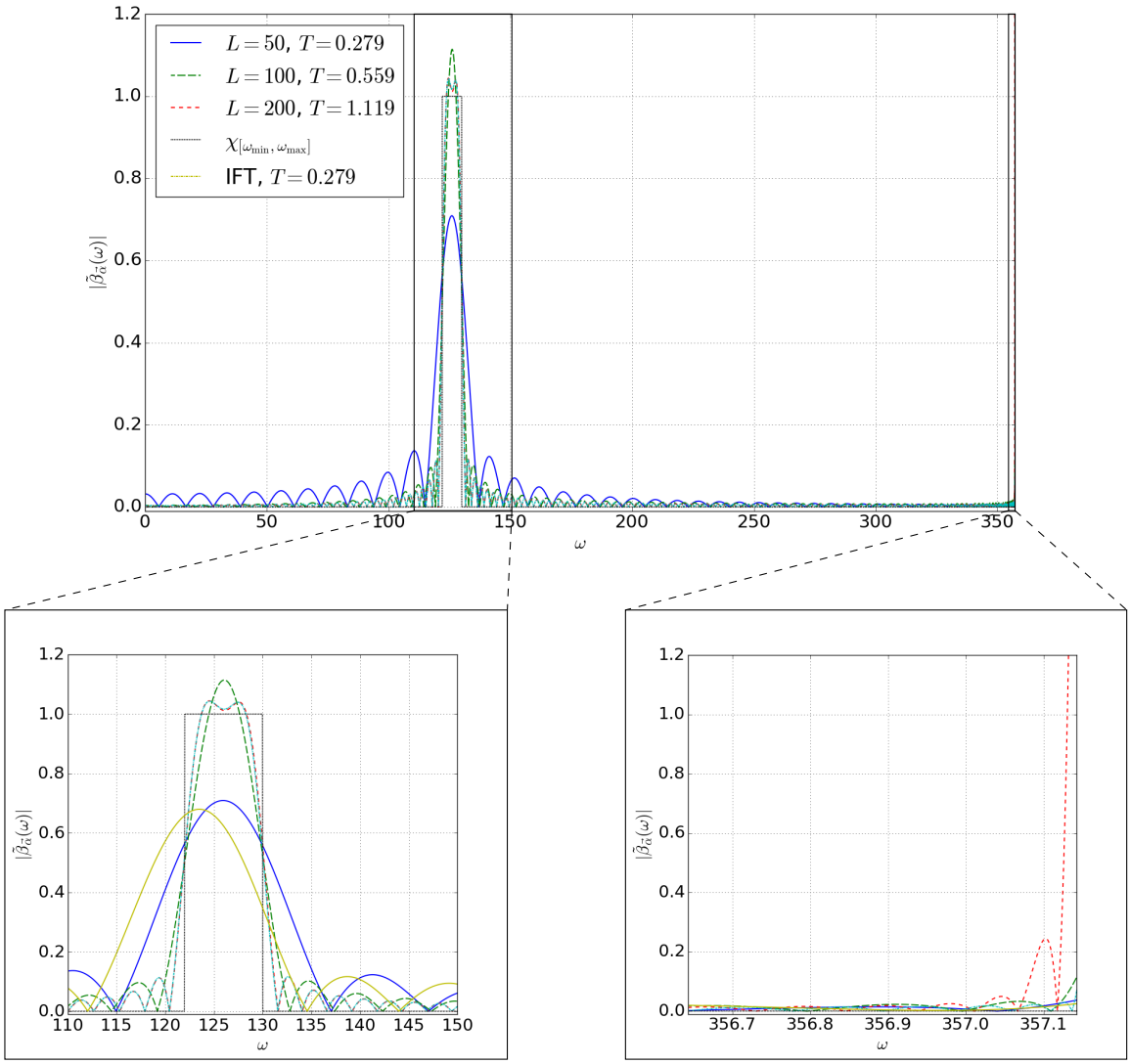


Figure 3.11: Plots of the function $\left|\tilde{\beta}_{\vec{\alpha}}\right|$ with the vector $\vec{\alpha}$ obtained by the L^2 minimization method with midpoint quadrature, $h = 0.05$ for the blue, red, and green curves, and $h = 0.001$ for the light blue curve. The target interval is $[\omega_{\min}, \omega_{\max}] = [122, 130]$, the time-step is $\tau = 0.0056$, and $\omega_{\text{end}} = 2/\tau \approx 357.14$. We vary the number of time-steps L. The yellow curve represents the discrete filter function obtained by the inverse Fourier transform of the indicator with end-time $T = 0.279$.

4 Numerical experiments

In this chapter we apply the deployed Krylov eigenvalue solver (Algorithm 4) to the problem stated in Definition 2.1 on a two-dimensional rectangular domain. We compare the convergence of the eigenvalues inside and outside of the target interval for different filter functions. For space discretization, computation of matrices M and S, and visualization of obtained eigenfunctions, we use the FEM library Netgen/NGSolve in Python.

We consider the domain $\Omega := \left[0, \sqrt[3]{2}\right] \times [0,1] \subset \mathbb{R}^2$ and fix a partition $\mathcal{T}$ with maximal size $h = 0.05$. As a discrete solution space, we choose first-order splines in $\mathbb{R}^2$: $V_h = \left\{v \in H^1(\Omega) : \quad \forall T \in \mathcal{T} : v|_T \in \mathbb{P}_1\right\}$. This leads to an $N = 629$-dimensional matrix eigenvalue problem.

The exact spectrum of the negative Laplace operator on Ω with Neumann boundary conditions is $\sigma(-\Delta) = \left\{\omega_{m,n}^2 := \left(m^2/\sqrt[3]{4} + n^2\right)\pi^2 : \quad m,n \in \mathbb{N}\right\}$. We expect some perturbations of the spectrum due to discretization in space; however, these errors do not depend on our method. Thus, we compare the convergence of eigenvalues obtained by the Krylov iteration to the eigenvalues of the N-dimensional matrix eigenvalue problem after partitioning. In other words, the reference spectrum $\sigma(M^{-1}S)$ consists of eigenvalues of the matrix $M^{-1}S$, denoted as $\tilde{\omega}_{m,n}^2$. We compute them numerically using the `numpy.linalg.eig()` function in Python. This method is also used as a direct solver to solve the projected m-dimensional problem.

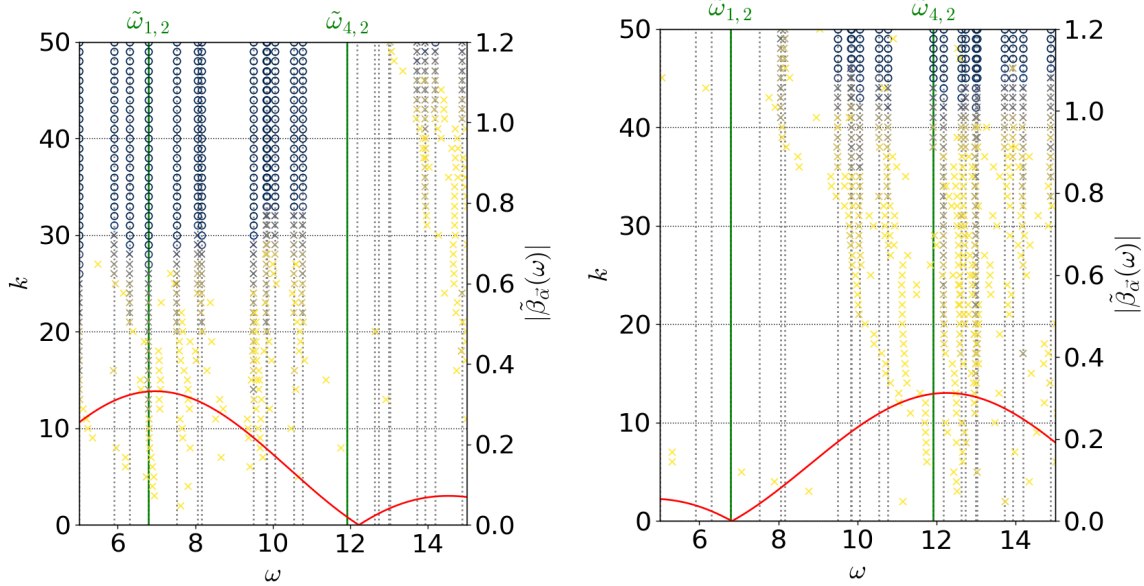
In the experiment, we select a time-step $\tau = 0.0056$, which corresponds to controlled interval between 0 and $\omega_{\text{end}} = 2/\tau \approx 357.14$, covering all resonances of the discrete problem 2.3. In each Krylov-step, we perform $L = 200$ time-steps. We construct the discrete filter function $\tilde{\beta}_{\vec{\alpha}}$ by the least squares fitting presented in Section 3.3 with $K = 1000$ Chebyshev nodes.

For each eigenvalue $\tilde{\omega}_{m,n}^2$ in the reference spectrum and each Krylov iteration, we denote with $\text{error}_k\left(\tilde{\omega}_{m,n}^2\right)$ the distance between ω_h^2 and the closest obtained eigenvalue $\omega_{(k)}^2$ in the k-th iteration, i.e.,

$$\text{error}_k\left(\tilde{\omega}_{m,n}^2\right) := \min\left\{\left|\omega_{(k)}^2 - \tilde{\omega}_{m,n}^2\right| : \quad \omega_{(k)}^2 \text{ is a result after the } k\text{-th iteration}\right\}.$$

Figure 4.1 presents the obtained resonance frequencies between 5 and 15 in the first 50 Krylov iterations for two different target intervals: $[\omega_{\min}, \omega_{\max}] = [6,8]$ and $[\omega_{\min}, \omega_{\max}] = [11,13]$. Furthermore, we compare results for $L = 100$ and $L = 200$ time-steps per iteration, which correspond to end-times $T = 0.56$ and $T = 1.12$, respectively. The left vertical axis represents the step of the Krylov iteration. Vertical dotted lines indicate reference resonances $\tilde{\omega}_{m,n}$, such that $\tilde{\omega}_{m,n}^2 \in \sigma(M^{-1}S)$. Plotted points are obtained resonances in each iteration. The color and mark of the point depend on $\text{error}_k\left(\tilde{\omega}_{m,n}^2\right)$: resonances with a small error are blue, and those with a larger error are yellow. Furthermore, points satisfying $\text{error}_k\left(\tilde{\omega}_{m,n}^2\right) < 10^{-5}$ are depicted with circles, while others are depicted with

$L = 100, T = 0.56$:



$L = 200, T = 1.12$:

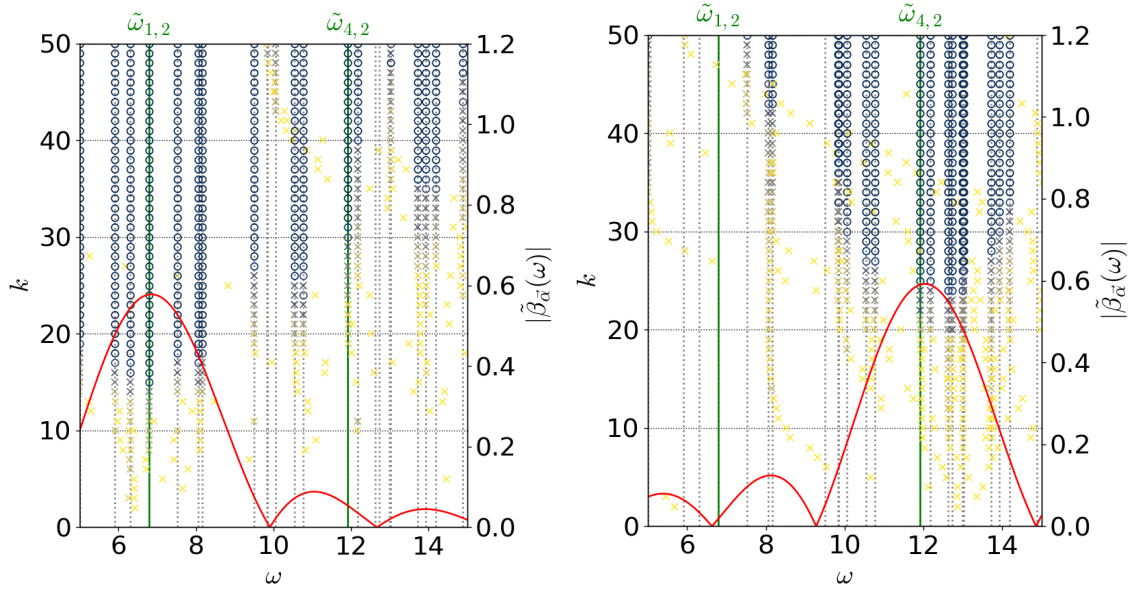


Figure 4.1: Convergence of the obtained resonances towards the reference ones in the first 50 Krylov iterations for $L = 100$ and $L = 200$ time-steps per iteration. The filter function is constructed using the least squares method with 1000 Chebyshev knots, and the target function $\chi_{[\omega_{\min},\omega_{\max}]}$ with target intervals $[6, 8]$ (right plots) and $[11, 13]$ (left plots).

crosses. The red curve (with values on the right vertical axis) represents the absolute value of the discrete filter function $\tilde{\beta}_{\vec{\alpha}}$.

We observe, that a smaller number of time-steps in each Krylov iteration slows down the convergence both inside and outside the target interval. Nevertheless, with $L = 100$ time-steps per iteration and in both target intervals, the method captured all sought eigenvalues in the region of interest after fewer than 50 iterations with a tolerance smaller than 10^{-5}. In other numerical experiments, we observed similar results.

For further error analysis, we focus on two particular resonances: $\omega_{1,2} \approx 6.76$ and $\omega_{4,2} \approx 11.78$, each in the middle of one of two target intervals. Figure 4.2 presents $\mathrm{error}_k\left(\tilde{\omega}^2_{1,2}\right)$ and $\mathrm{error}_k\left(\tilde{\omega}^2_{4,2}\right)$ depending on the step k and filter function. Note that the approximation of the resonance obtained by the Krylov iteration cannot worsen, since in each iteration the algorithm enlarges the Krylov space. If an eigenvector v lies in the subspace spanned by B_k, then it also lies in the subspace spanned by B_{k+1}. Therefore, we truncate these plots to 25 and 35 steps, respectively, where the approached tolerance is much smaller than the tolerance resulting from the space-discretization, and further iteration does not impact this particular resonance.

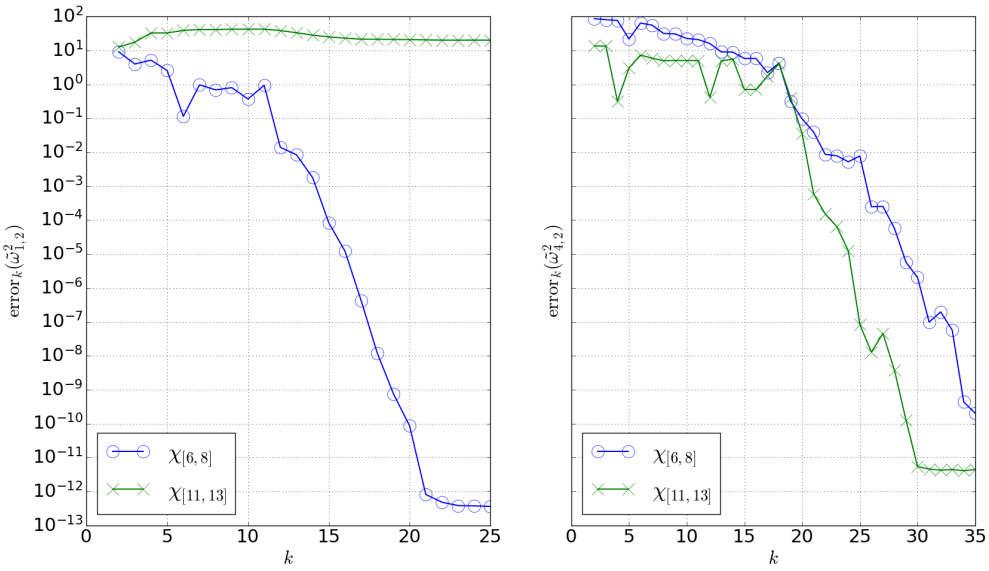


Figure 4.2: Error between perturbed eigenvalues $\tilde{\omega}^2_{1,2}$ (left plot) and $\tilde{\omega}^2_{4,2}$ (right plot) for target functions $\chi_{[6,8]}$ (blue line) and $\chi_{[11,13]}$ (green line). $L = 200$ time-steps per iteration.

We observe much faster convergence of the method in regions, where $\left|\tilde{\beta}_{\vec{\alpha}}\right|$ is larger. Especially, in the middle of the target interval the method with parameter $L = 200$ yields accuracy in ω^2 exceeding 10^{-10} after fewer than 30 steps. On the other hand, resonances outside the target interval, where the filter function is closer to 0, converge much slower or seem not to be captured in the first 50 iterations at all. In other numerical experiments

with different filter functions, we also observed a similarly strong correlation between the absolute amplitude of the discrete filter function and the convergence rate of the eigenvalues.

In the carried out experiments, we reduced the size of the problem from $N = 629$ to $m = 20$ or $m = 50$ (performing $L = 200$ or $L = 100$ time-steps per iteration, respectively) and obtained all eigenvalues in the target intervals with precision exceeding 10^{-5} relative to the reference spectrum.

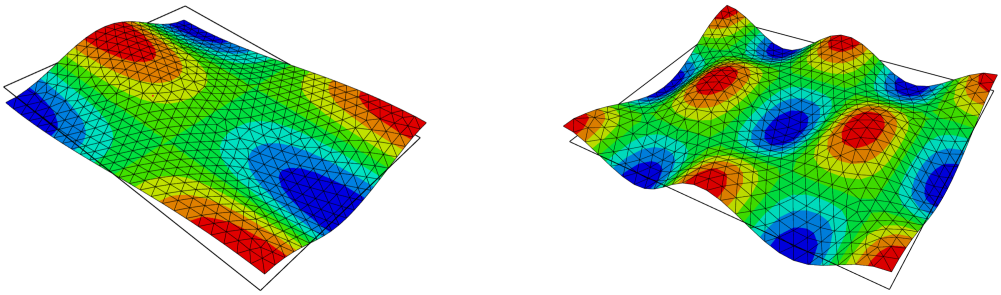


Figure 4.3: Eigenfunctions corresponding to resonances $\omega_{1,2}$ (left plot) and $\omega_{4,2}$ (right plot) obtained after 20 steps of Krylov iteration with appropriate target intervals. The exact solutions are: $u_{m,n}(x, y) = \cos\left(m\pi x/\sqrt[3]{2}\right)\cos(n\pi y)$.

5 Conclusions

In this thesis, we presented a method to compute eigenvalues of the high-dimensional operators. Similar to [NW24], we focused on the negative Laplace operator in $\mathbb{R}^2$ and $\mathbb{R}^3$, since it plays a significant role in the theory of wave equations. Nevertheless, the deployed algorithm can be easily reformulated for different problems, as long as the discretization matrices S and M are symmetric and positive (semi-)definite.

We introduced the operator C as the discrete equivalent of the weighted time-integral of the solution to the wave equation (2.5) with the weight function α. The method projects the high-dimensional problem onto the Krylov space of this operator, significantly reducing the dimension of the matrix eigenvalue problem to solve.

A crucial role in the construction of the operator C is played by the weight function α (or the weight vector $\vec{\alpha}$ due to replacing the integral by quadrature rule) and corresponding discrete filter function $\tilde{\beta}_\alpha$. Lemma 2.23 quantified the correspondence between eigenpairs of the operator C and the sought eigenpairs of the negative Laplace operator. Using an appropriate filter function, the method can filter out resonances outside the desired region of interest, so that the solution of the projected problem consists mostly of the sought eigenvalues.

This thesis focuses on the suitable choice of the weights. We compared different approaches to constructing of the vector $\vec{\alpha}$. As the first method, we followed the choice in [NW24], using the scaled inverse Fourier transform of the goal function truncated to a finite time-interval (forced by the time-stepping in each Krylov step).

Since the discrete filter function $\tilde{\beta}_{\vec{\alpha}}$ is a polynomial in ω^2, a natural approach was to select collocation nodes in the controlled interval and solve the collocation problem. Due to the conditioning of the obtained linear system of equations, we used the Chebyshev knots in ω^2. However, because of the non-uniform distribution of the knots in the controlled interval and higher density at the end of the interval, it is impossible to precisely control the filter function, especially with a lower parameter L, which determines the degree of the polynomial (and thus the number of collocation nodes), but also impacts the computation costs of the algorithm. In special cases, if no collocation node lies in the target interval, the method yields a constant zero function, which is not acceptable.

A remedy to this disadvantage of the method is to employ more sample points in the controlled interval and use the least squares fitting instead of the collocation. The problem is well-conditioned independently of the degree of the polynomial by sampling in the Chebyshev knots; with other distributions of nodes, the conditioning can be controlled by increasing the number of nodes. Furthermore, we have proven, that the least squares method with equidistant nodes is equivalent to the fitting with respect to the L^2 norm (with the integral replaced by the rectangle rule). In the limit, by increasing number of nodes, the solutions given by the least squares method converge towards the L^2 error minimum.

In most cases, we did not observe significant differences between the presented methods.

When the target function is the indicator function of the interval that is closer to 0 than to ω_{end} (which is mostly the case in applications), the simplest way to construct the weight function remains the inverse Fourier transform method presented in Section 3.1. In some settings, this method displays undesirable behavior at the end of the controlled interval, or the peak of the function seems to be shifted from the target interval. Thus, in such cases, the best results were yielded by the least squares fitting in Chebyshev knots in ω^2. It is simple to adjust the number of nodes to avoid having of the target interval between two adjacent knots. Due to the good conditioning of the problem, this method results in an accurate filter function, even at the end of the controlled interval. Higher computational costs of this method are negligible, since the computation of the weights is a preprocessing step and is not part of the algorithm.

In the experiments carried out in Chapter 4, we computed the eigenvalues of the negative Laplace operator on the rectangular domain in $\mathbb{R}^2$ in two target intervals. Thanks to the projection onto the Krylov space obtained by an appropriate filter function, we significantly reduced the size of the problem, and therefore also the computational costs, without loss of accuracy in the obtained eigenpairs.

Bibliography

[INPR23] Michael Innerberger, Lothar Nannen, Dirk Praetorius, and Alexander Rieder. Numerics of differential equations. Lecture notes, TU Wien, February 2023.

[Jü23] Ansgar Jüngel. Partielle Differentialgleichungen. Lecture notes, TU Wien, August 2023.

[Nan23] Lothar Nannen. Numerische Mathematik A. Lecture notes, TU Wien, August 2023.

[NW24] Lothar Nannen and Markus Wess. A Krylov eigenvalue solver based on filtered time domain solutions, 2024.

[Pra22] Dirk Praetorius. Numerische Mathematik (WS 2022/23). Lecture notes, TU Wien, 2022.