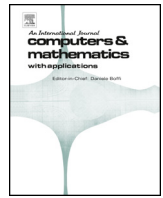




Contents lists available at ScienceDirect

## Computers and Mathematics with Applications

journal homepage: [www.elsevier.com/locate/camwa](http://www.elsevier.com/locate/camwa)

## A Krylov eigenvalue solver based on filtered time domain solutions

Lothar Nannen, Markus Wess\*

Institute of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040, Vienna, Austria

## ARTICLE INFO

## Keywords:

Eigenvalue solver  
 Filtered Krylov iteration  
 Acoustic resonance  
 Explicit time-stepping

## ABSTRACT

This paper introduces a method for computing eigenvalues and eigenvectors of a generalized Hermitian, matrix eigenvalue problem. The work is focused on large scale eigenvalue problems, where the application of a direct inverse is out of reach. Instead, an explicit time-domain integrator for the corresponding wave problem is combined with a proper filtering and a Krylov iteration in order to solve for eigenvalues within a given region of interest. We report results of small scale model problems to confirm the reliability of the method, as well as the computation of acoustic resonances in a three dimensional model of a hunting horn to demonstrate the efficiency.

## 1. Introduction

Computing eigenvalues and eigenvectors of large scale eigenvalue problems is still a challenging task in applied mathematics. We consider in this paper the computation of eigenpairs  $(\omega^2, v)$  to the generalized matrix eigenvalue problem  $Sv = \omega^2 Mv$  with sparse Hermitian, positive (semi-)definite matrices  $S$  and  $M$  generated by a finite element discretization of a Laplacian eigenvalue problem. Even in this most simple setting numerical solvers struggle if the matrices become large and if non-extremal or clustered eigenvalues are sought.

We refer to [1] for the standard algorithms for such problems. All of the methods therein rely on the basic principle of a simple power iteration which might be combined e.g., with a Krylov subspace method. Unfortunately, with this basic idea only eigenvalues with largest absolute value can be computed efficiently. One remedy is the use of inverse or shift-and-inverse iterations, which allow in general the computation/approximation of eigenvalues with smallest absolute value or closest to a chosen shift parameter. The price to pay is the need to invert a large, sometimes indefinite matrix in each iteration step. If the matrix dimension is too large for a direct solver to be feasible, then in each step iterative solvers have to be used leading to very high computation times and/or the need for efficient preconditioners.

There exist several variants of these methods like the filtered subspace iteration (FEAST), see [2,3], or the contour integral method introduced in [4,5] for eigenvalue problems, which are non-linear in the eigenvalue. These methods can reduce the number of iterations, since they focus the iterations to the sought eigenvalues similar to shift-and-inverts described above. Again in each iteration step of these methods

several linear systems of equations have to be solved. Hence these methods are efficient as long as a direct solver can be used.

The locally optimal block preconditioned conjugate gradient method (LOBPCG), see [6], avoids the solution of large linear systems of equations with the additional requirement of a preconditioner for the matrix  $S$ . With this method, even for large systems, the smallest eigenvalues can be computed efficiently. However, we also consider situations where not only the smallest eigenvalues but also eigenvalues within a certain interval are of interest. For the latter, the LOBPCG method cannot be used in a straightforward way.

In this paper we propose a method, which is related to time domain solvers for the Helmholtz problem, see [7–9]. We construct a Krylov space based on filtered time domain solutions. More precisely, in each Krylov iteration several steps of an explicit time-stepping scheme are applied to a semi-discrete wave equation. The time domain solutions are then combined with a proper weight function in order to construct the next Krylov vector. Finally, the large matrix eigenvalue problem is projected and solved on the small dimensional Krylov space. The crucial part of the method is the choice of the weight function. We construct a discrete weight function based on an inverse Fourier transform of a characteristic function of an interval of interest, where the eigenvalues are sought.

We show that this method results in a polynomial filtering for the matrix  $M^{-1}S$ . Even though polynomial filters are more popular for the computation of eigenvalues with largest magnitude they can be used for non-extremal eigenvalues as well (see e.g., [10]). In contrast to the filter constructed in [10], our approach is motivated by the underlying partial differential equation. We exploit the efficiency and flexibility

\* Corresponding author.

E-mail addresses: [lothar.nannen@tuwien.ac.at](mailto:lothar.nannen@tuwien.ac.at) (L. Nannen), [markus.wess@tuwien.ac.at](mailto:markus.wess@tuwien.ac.at) (M. Wess).

of already existing explicit solvers for wave equations. In particular, we use mass lumping techniques (see [11]) for a cheap inversion of the mass matrix. Moreover, choosing stable time-domain discretizations automatically yields polynomial filters fitted to the numerical range of the matrix eigenvalue problem.

The remainder of the paper is organized as follows. The basic concept of the method is presented in Section 2. Section 3 contains numerical examples for a small scale as well as a large scale problem. The small scale problem is used to give hints for choosing suitable method parameters. In the large scale problem we compute resonance frequencies of a three dimensional model of a hunting horn in a closed room. The eigenvalues, which model the playable notes on the horn belong to a region where the background eigenvalues of the room are already quite dense, such that the LOBPCG method is not feasible anymore. The same holds true for methods using a direct solver, since the application of such a solver to the systems with more than  $10^6$  unknowns typically exceeds the computer memory of a standard desktop computer. The paper closes with a discussion of extensions to the method in Section 4 and a short conclusion in Section 5.

## 2. Presentation of the method

In this section we explain the concept in a most simple setting. For a bounded Lipschitz domain  $\Omega \subset \mathbb{R}^d$  (for  $d = 2, 3$ ) we solve for eigenpairs  $(\omega^2, u)$  with  $\omega \geq 0$  and non-trivial  $u \in H^1(\Omega)$  of the negative Neumann-Laplacian, i.e.,  $(\omega^2, u)$  solves

$$-\Delta u = \omega^2 u \quad \text{in } \Omega, \tag{1a}$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{at } \partial\Omega. \tag{1b}$$

In the following, we will refer to  $\omega$  as resonance or resonance frequency, if  $\omega^2$  is an eigenvalue.

The problem is discretized using a standard Galerkin method. We choose a partition  $\mathcal{T}$  of  $\Omega$  consisting of simplexes and use the discrete finite element space

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

where  $\mathcal{P}_p$  denotes the space of polynomials up to degree  $p \in \mathbb{N}$ . Thus  $V_h$  consists of piecewise polynomials. We solve for discrete eigenpairs  $(\omega_h^2, u_h)$  with  $\omega_h \geq 0$  and non-trivial  $u_h \in V_h$  of the variational formulation

$$\int_{\Omega} \nabla u_h \cdot \nabla \varphi \, dx = \omega_h^2 \int_{\Omega} u_h \varphi \, dx \quad \forall \varphi \in V_h.$$

In the following, we omit the index  $h$  since we fix the spatial discretization and focus on the equivalent matrix eigenvalue problem to find eigenvectors  $v \in \mathbb{R}^N \setminus \{0\}$  and eigenvalues  $\omega^2 \geq 0$  with  $N := \dim V_h \in \mathbb{N}$  such that

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

The self-adjoint matrices  $S = (s_{ij})$ ,  $M = (m_{ij})$  are obtained by choosing basis functions  $\psi_n$  such that  $V_h = \text{span}\{\varphi_1, \dots, \varphi_N\}$  with

$$s_{ij} := \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j \, dx, \quad m_{ij} := \int_{\Omega} \varphi_i \varphi_j \, dx, \quad i, j = 1, \dots, N.$$

Clearly,  $M$  is positive definite and  $S$  positive semi-definite.

**Remark 2.1.** Due to the fact that in the following it will become necessary to apply the inverse of the mass matrix  $M$ , instead of evaluating the integrals exactly one may approximate them by numerical integration such that the sparsity pattern of  $M$  is more favorable (e.g.,  $M$  is diagonal if mass lumping is used, see [11]). See Sec. 4 for other discretizations leading to a cheap inverse of the mass matrix.

### 2.1. Krylov eigenvalue solver

Let us assume that there exists a matrix  $C \in \mathbb{R}^{N \times N}$  such that the eigenvectors of the matrix eigenvalue problem (2) are also eigenvectors of the auxiliary eigenvalue problem

$$Cw = \mu w. \tag{3}$$

This includes the possibility that an eigenspace of (3) corresponding to a single eigenvalue  $\mu$  is the direct sum of two or more eigenspaces of (2) corresponding to different eigenvalues (cf., also Lemma 2.3 and Remark 2.2). Moreover, let

$$\mathcal{K}_m(C, r_0) := \text{span}\{r_0, Cr_0, \dots, C^{m-1}r_0\},$$

be the Krylov space of  $C$  with a normalized random starting vector  $r_0 \in \mathbb{R}^N$  and  $m \in \mathbb{N}$ . An orthonormal basis  $\{b_0, \dots, b_{m-1}\}$  of  $\mathcal{K}_m(C, r_0)$  can be computed iteratively using a Gram-Schmidt orthogonalization, i.e., using  $b_0 = r_0$  and for each  $j = 1, \dots, m-1$  we compute  $Cb_{j-1}$  and orthonormalize it with respect to  $b_0, \dots, b_{j-1}$ . Using the projection matrix  $B_m = (b_0 \dots b_{m-1}) \in \mathbb{R}^{N \times m}$  we project the original eigenvalue problem (2) onto the Krylov space generated by the matrix  $C$ : find eigenpairs  $(\omega_m^2, v_m) \in \mathbb{R} \times \mathbb{R}^m \setminus \{0\}$  such that

$$B_m^T S B_m v_m = \omega_m^2 B_m^T M B_m v_m. \tag{4}$$

Typically,  $m$  is small compared to  $N$ . Hence, the  $m$ -dimensional eigenvalue problem (4) can be solved with low computational costs.

**Remark 2.2.** An optimal filter choice of the matrix  $C$  would be such that the eigenvectors corresponding to sought eigenvalues form the eigenspace of  $C$  of the eigenvalue  $\mu = 1$  and all the remaining eigenvectors lie in the kernel of  $C$ . In this case the algorithm yields the exact sought eigenpairs after  $l$  Krylov steps, where  $l$  is the number of the sought eigenvalues.

Since (2) is a Hermitian eigenvalue problem, following [1, Theorem 4.6 and Sec. 6.7] we expect convergence of the eigenvectors of (4) towards the eigenspaces to the eigenvalues  $\mu$  of (3) with largest absolute values. Hence, the projected eigenvalues  $\omega_m^2$  converge towards those eigenvalues  $\omega^2$  of the original problem, for which the corresponding eigenvalues  $\mu$  of the auxiliary problem (3) have largest absolute value.

Following the considerations above the matrix  $C$  has to be chosen in a way that the eigenvalues  $\mu$  of (3) corresponding to the eigenvalues  $\omega^2$  of interest of (2) have large absolute values, while the remaining eigenvalues of (3) are close to zero. Most standard would be to choose a shift parameter  $\rho$  such that  $S - \rho M$  is regular and use the shift-and-invert matrix  $C := (S - \rho M)^{-1} M$ . It is straightforward to show, that with this choice of  $C$  the eigenvectors of (2) are identical to the ones of (3) and that the correspondence of the eigenvalues is given by  $\omega^2 = \rho + \mu^{-1}$ . Hence, the presented method yields approximations to the squared eigenvalues closest to the shift parameter  $\rho$ . Unfortunately, using this shift-and-invert technique requires the application of the inverse  $(S - \rho M)^{-1}$ . In other words, in each Krylov step a linear system of equations for the matrix  $S - \rho M$  of dimension  $N$  has to be solved. For problems small enough such that a direct solver can be used efficiently, this shift-and-invert approach can be applied with reasonable computational costs.

However, in this paper we are interested in problems where the application of a direct solver is out of reach. An alternative for a direct inverse would be the use of iterative solvers. For Helmholtz type problems in resonating structures the use of such solvers can be challenging. Thus we choose a different approach and define the operator  $C$  based on filtered time-domain solutions of the underlying wave problem.

Note that our approach differs from applying the Arnoldi method directly to the auxiliary problem (3), since we project the original problem (2) onto the Krylov space constructed by the auxiliary problem (3).

In our experiments it turned out that a stopping criterion for the Krylov iterations is easier to construct for the projected original problem (4). Moreover, we are interested in  $\omega$  and not the auxiliary eigenvalues  $\mu$ . For a shift-and-invert method the mapping  $\omega^2 \mapsto \mu$  is one-to-one, i.e.  $\omega$  can be easily computed if  $\mu$  is known. This is not the case for the method based on filtered time-domain solutions.

2.2. Filtered time-domain solutions

To motivate our specific choice of the auxiliary problem (3), we follow the approach from [8], which was developed for scattering problems. For given  $r \in \mathbb{R}^N$  let  $y(\cdot; r) : [0, \infty) \rightarrow \mathbb{R}^N$  be the solution to the semi-discrete wave problem

$$M\ddot{y}(t; r) = -Sy(t; r), \quad \text{for } t > 0, \tag{5a}$$

$$y(0; r) = r, \quad \dot{y}(0; r) = 0, \tag{5b}$$

where  $\dot{y}(\cdot; r)$  and  $\ddot{y}(\cdot; r)$  denote the first and second time derivative. Since  $M, S$  are Hermitian matrices, there exists an orthonormal basis of eigenvectors  $v_j \in \mathbb{R}^N, j = 1, \dots, N$  with corresponding eigenvalues  $\omega_j^2$  to (2) and the unique solution to (5) is given by

$$y(t; r) = \sum_{j=1}^N \cos(\omega_j t) (v_j^\top r) v_j. \tag{6}$$

For a given piecewise continuous weight function  $\alpha : [0, \infty) \rightarrow \mathbb{R}$  with compact support we define the integral operator  $\Pi_\alpha : \mathbb{R}^N \rightarrow \mathbb{R}^N$  by

$$\Pi_\alpha r := \int_0^\infty \alpha(t) y(t; r) dt. \tag{7}$$

A discrete version of this integral operator will adopt the role of the matrix  $C$  from the preceding subsection. The following Lemma quantifies the correspondence of eigenpairs of the initial matrix eigenvalue problem (2) and the ones of  $\Pi_\alpha$ .

**Lemma 2.3.** *Let  $(\omega^2, v)$  be an eigenpair of (2) and the filter function  $\beta_\alpha : [0, \infty) \rightarrow \mathbb{R}$  be defined by*

$$\beta_\alpha(s) := \int_0^\infty \alpha(t) \cos(ts) dt. \tag{8}$$

*Then  $(\beta_\alpha(\omega), v)$  is an eigenpair of  $\Pi_\alpha$ , i.e.,  $\Pi_\alpha v = \beta_\alpha(\omega)v$ . Vice versa, if  $(\lambda, v)$  is an eigenpair of  $\Pi_\alpha$ , then there exists at least one eigenvalue  $\omega^2$  of (2) such that  $\beta_\alpha(\omega) = \lambda$  and  $v$  belongs to the sum of eigenspaces of those eigenvalues  $\omega^2$  of (2) for which  $\beta_\alpha(\omega) = \lambda$ .*

**Proof.** Its straightforward to show, that the solution  $y(\cdot; v)$  of (5) is given by  $y(t; v) = \cos(\omega t)v$  if  $(\omega^2, v)$  is an eigenpair of (2). Hence, the first claim holds by definition of  $\beta_\alpha$  and  $\Pi_\alpha$ . If  $(\lambda, v)$  is an eigenpair of  $\Pi_\alpha$ , then the representation (6) of the solution to the wave equation yields

$$\left( \int_0^\infty \alpha(t) \cos(\omega_j t) dt - \lambda \right) v_j^\top v = 0, \quad j = 1, \dots, N,$$

since the eigenvectors  $v_j$  form an orthonormal basis of  $\mathbb{R}^N$ . For  $\omega_j$  with  $\lambda \neq \beta_\alpha(\omega_j)$  this implies  $v_j^\top v = 0$ . Since  $v \neq 0$ , there exists at least one  $\omega$  with  $\lambda = \beta_\alpha(\omega)$ .  $\square$

Lemma 2.3 shows that eigenvectors of the original eigenvalue problem are also eigenvectors of the operator  $\Pi_\alpha$ . In particular, if we use  $\Pi_\alpha$  to construct a Krylov space, this Krylov space approximates sums of eigenspaces to the original problem only. If this were not the case Krylov steps would possibly be wasted into approximations of eigenvectors to  $\Pi_\alpha$  which are irrelevant for the original eigenvalue problem.

In what follows, we motivate our choice of the weight function  $\alpha$ . Using the symmetric extension  $\hat{\alpha}(t) := \alpha(-t)$  for  $t < 0$ , the function  $\beta_\alpha$  can be represented by the Fourier transform of  $\hat{\alpha}$ :

$$\begin{aligned} \beta_\alpha(s) &= \int_0^\infty \alpha(t) \cos(ts) dt \\ &= \frac{1}{2} \int_{-\infty}^\infty \hat{\alpha}(t) \exp(-its) dt = \sqrt{\frac{\pi}{2}} \mathcal{F}(\hat{\alpha})(s). \end{aligned}$$

Since the presented Krylov method converges towards the eigenvalues with largest absolute values, the weight function  $\alpha$  should be adapted to the location of the sought eigenvalues. E.g., if eigenvalues  $\omega$  in an interval  $[\omega_-, \omega_+]$  are sought, it would be optimal to find  $\alpha$  such that  $\beta_\alpha = \chi_{[\omega_-, \omega_+]}$  where  $\chi_A$  denotes the characteristic function of a set  $A$ . This would correspond to the choice of  $\alpha$  as

$$\sqrt{\frac{2}{\pi}} \mathcal{F}^{-1} \left( \chi_{[\omega_-, \omega_+]} \right) = \frac{4}{\pi t} \sin \left( \frac{t}{2} (\omega_+ - \omega_-) \right) \cos \left( \frac{t}{2} (\omega_+ + \omega_-) \right), \tag{9}$$

for  $t > 0$ . Obviously, the function on the right hand side is not compactly supported in  $[0, \infty)$ . However we are not forced to use exactly this weight function. In fact, in the following we choose a finite time interval  $[0, T]$  with  $T > 0$  together with the weight function

$$\alpha(t) := \begin{cases} \frac{2(\omega_+ - \omega_-)}{\pi}, & t = 0, \\ \frac{4}{\pi t} \sin \left( \frac{t}{2} (\omega_+ - \omega_-) \right) \cos \left( \frac{t}{2} (\omega_+ + \omega_-) \right), & t \in (0, T], \\ 0, & t > T. \end{cases} \tag{10}$$

Note that the method is not limited to this specific weight function (see also Section 4.3). In general, for fast convergence the function  $\alpha$  should be chosen such that  $|\beta_\alpha(\omega)| \gg |\beta_\alpha(\tilde{\omega})|$  for the sought  $\omega$  and the unsought  $\tilde{\omega}$ .

2.3. Discretization of the filtered time-domain solution

We discretize the integral operator (7) using a rectangle rule. To this end, we introduce a time-stepping method for the wave equation (5). Since we are interested in eigenvalue problems where applying a direct solve is out of reach we focus on explicit methods. For a fixed, uniform step-size  $\tau > 0$ , which is specified later, we approximate  $y(\ell \tau; r)$  for  $\ell \in \mathbb{N}$  by  $y_\ell(r)$  using finite differences. This leads to the explicit, second order two-step method

$$y_{\ell+1}(r) = -\tau^2 M^{-1} S y_\ell(r) + 2y_\ell(r) - y_{\ell-1}(r), \quad \ell \in \mathbb{N}, \tag{11}$$

with initial time steps

$$y_{-1}(r) = y_0(r) = r. \tag{12}$$

Note, that we define  $y_{-1}(r) = r$  since  $y_{-1}(r)$  approximates  $y(-\tau; r) = y(0; r) - \tau \dot{y}(0; r) + \mathcal{O}(\tau^2) = r + \mathcal{O}(\tau^2)$ .

Using the rectangle rule with  $L \in \mathbb{N}$  quadrature points and therefore the step-size  $\tau := T/L$ , we finally arrive at the fully discrete linear mapping  $C : \mathbb{R}^N \rightarrow \mathbb{R}^N$ :

$$r_{j-1} \mapsto r_j = C r_{j-1} := \sum_{\ell=0}^{L-1} \tau \alpha(\ell \tau) y_\ell(r_{j-1}), \tag{13}$$

which approximates the integral operator (7).

In order to find the discrete analogue to the filter function  $\beta_\alpha$  defined in (8) we study the scalar linear difference problems: for  $\omega > 0$  let  $q_\ell(\omega) \in \mathbb{R}$  for  $\ell \in \mathbb{N}_0$  be the unique solution to

$$q_{-1}(\omega) = q_0(\omega) = 1, \quad q_{\ell+1}(\omega) = (2 - \tau^2 \omega^2) q_\ell(\omega) - q_{\ell-1}(\omega). \tag{14}$$

Note, that diagonalization of the matrix  $M^{-1} S$  separates (11) into such problems for those  $\omega > 0$  such that  $\omega^2$  are eigenvalues of the matrix eigenvalue problem (2). It is straightforward to show that

$$q_\ell(\omega) = \frac{1 - \overline{c(\omega)}}{2i\Im(c(\omega))} c(\omega)^\ell + \frac{c(\omega) - 1}{2i\Im(c(\omega))} \overline{c(\omega)}^\ell$$

$$= \frac{\Im(c(\omega)^\ell) - |c(\omega)|^2 \Im(c(\omega))^{\ell-1}}{\Im(c(\omega))},$$

with

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

Hence,  $|c(\omega)| = 1$  if the CFL condition  $\tau < 2/\omega$  is satisfied. In this case  $|q_\ell(\omega)|$  is bounded in  $\ell$ . Otherwise the sequence  $q_\ell(\omega)$  is unbounded for  $\ell \rightarrow \infty$ .

**Remark 2.4.** The time-stepping is stable if the step-size  $\tau$  is smaller than  $2/\omega$  for all eigenvalues  $\omega^2$  of  $M^{-1}S$ . For typical triangular finite element discretizations it can be shown that the maximal eigenvalues  $\omega_{\max}^2$  of  $M^{-1}S$  are related to the smallest mesh-size  $h$  of the used triangulation, i.e.,  $0 \leq \omega^2 < C/h^2$  with a constant  $C > 0$ . Hence, up to a constant, the time step-size  $\tau$  needs to be smaller than the spatial mesh-size  $h$ .

In practice we determine the maximal eigenvalue of  $M^{-1}S$  using a simple power iteration and choose the maximal step-size  $\tau$  such that the time-stepping is stable. Note, that smaller values of  $\tau$  lead to larger numbers of time-steps  $L$  to reach the end-time  $T = L\tau$  used in the weight function (10).

**Lemma 2.5.** The fully discrete filter function  $\tilde{\beta}_\alpha : [0, \infty) \rightarrow \mathbb{R}$  defined by

$$\omega \mapsto \tilde{\beta}_\alpha(\omega) := \sum_{\ell=0}^{L-1} \tau \alpha(\tau^\ell) q_\ell(\omega), \tag{15}$$

is a polynomial of degree  $L - 1$  in  $\omega^2$  and there holds

$$C = \tilde{\beta}_\alpha \left( \sqrt{M^{-1}S} \right). \tag{16}$$

In particular, if  $(\omega^2, v)$  is an eigenpair of (2), then  $(\tilde{\beta}_\alpha(\omega), v)$  is an eigenpair of  $C$ , and if  $(\lambda, v)$  is an eigenpair of  $C$  then there exists at least one eigenvalue  $\omega^2$  of (2) with  $\lambda = \tilde{\beta}_\alpha(\omega)$  and  $v$  belongs to the sum of eigenspaces of those eigenvalues  $\omega^2$  of (2) for which  $\tilde{\beta}_\alpha(\omega) = \lambda$ .

**Proof.** By definition of the scalar time-stepping (14)  $q_\ell(\omega)$  is a polynomial of degree  $\ell$  in  $\omega^2$ . Hence,  $\tilde{\beta}_\alpha(\omega)$  defined in (15) is a polynomial of degree  $L - 1$  in  $\omega^2$ . To prove (16) we start with an arbitrary vector  $r$  and the orthonormal basis of eigenvectors  $v_j$  to (2) corresponding to the eigenvalues  $\omega_j^2$ . The solutions  $y_\ell(r)$  to (11) are given by

$$y_\ell(r) = y_\ell \left( \sum_{j=1}^N (r^\top v_j) v_j \right)$$

$$= \sum_{j=1}^N (r^\top v_j) y_\ell(v_j) = \sum_{j=1}^N (r^\top v_j) q_\ell(\omega_j) v_j.$$

Hence, by definition of the matrix application  $C$  in (13) there holds

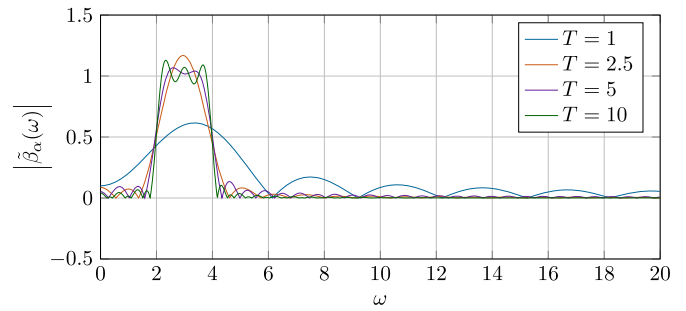
$$Cr = \sum_{\ell=0}^{L-1} \tau \alpha(\tau^\ell) \sum_{j=1}^N (r^\top v_j) q_\ell(\omega_j) v_j = \sum_{j=1}^N \tilde{\beta}_\alpha(\omega_j) (r^\top v_j) v_j$$

$$= V \text{diag}(\tilde{\beta}_\alpha(\omega_1), \dots, \tilde{\beta}_\alpha(\omega_N)) V^\top r = \tilde{\beta}_\alpha \left( \sqrt{M^{-1}S} \right) r,$$

where we used the spectral decomposition  $M^{-1}S = V^\top \text{diag}(\omega_1, \dots, \omega_N) V$  with  $V := (v_1, \dots, v_N)$ . The equivalence of eigenpairs of (2) to eigenpairs of  $C$  is a direct consequence of (16).  $\square$

In other words, we use a specific type of polynomial filtering.

**Remark 2.6.** Typically, polynomial filters are used for the computation of maximal eigenvalues (see [1]), but they can be used for the compu-



**Fig. 1.** Discrete filter functions for fixed time-step  $\tau = 0.025$ , the target interval  $[\omega_-, \omega_+] = [2, 4]$ , and varying end times  $T$ .

tation of non-extremal eigenvalues as well [10]. Instead of working on the linear algebra level we use an existing, stable solver for the wave equation. This solver can be exchanged easily by a different one as long as the time-stepping is diagonalized by the eigenvectors  $v_j$ , i.e., if  $y_\ell(v_j) = f_\ell(\omega_j) v_j$  holds for some functions  $f_\ell$ .

Fig. 1 shows discrete filter functions  $\tilde{\beta}_\alpha$  of (15) for fixed time-step  $\tau$  and  $[\omega_-, \omega_+] = [2, 4]$  in (9), while varying the end time  $T$  (and thus also the number of total time-steps). While the filter certainly separates the wanted from the unwanted eigenvalues better for larger end times  $T$ , the total number of necessary time-steps in the Arnoldi process can not be predicted by solely looking at the discrete filter function (see numerical experiments in Section 3).

#### 2.4. Algorithm

The considerations above lead to the following basic algorithm for computing the Krylov subspace  $\mathcal{K}_m(C, r_0)$ .

**Algorithm 2.7** (Krylov subspace by filtered time-domain solutions).

**Input:** matrices  $M^{-1}, S \in \mathbb{R}^{N \times N}$ , random starting vector  $r_0 \in \mathbb{R}^N$  with  $\|r_0\|_2 = 1$ , number of time steps  $L \in \mathbb{N}$ , dimension of Krylov space  $m \in \mathbb{N}$ , weight function  $\alpha$

- 1: find max. stable step-size  $\tau$  by power iteration for  $M^{-1}S$
- 2: **for**  $k = 0, \dots, m - 1$  **do** Krylov loop
- 3:    $y_0 := y_1 := r_k$
- 4:    $r_{k+1} := \tau \alpha(0) y_0$  first term in (13)
- 5:   **for**  $i = 1, \dots, L - 1$  **do** time loop
- 6:      $y_2 := -\tau^2 M^{-1}S y_1 + 2y_1 - y_0$  time step (11)
- 7:      $r_{k+1} += \tau \alpha(i\tau) y_2$  additions in (13)
- 8:      $y_0 := y_1, y_1 := y_2$
- 9:   **end for**
- 10:   **for**  $j = 0, \dots, k$  **do** orthogonalization loop
- 11:      $r_{k+1} -= (r_{k+1}, r_j)_2 r_j$
- 12:   **end for**
- 13:   **if**  $\|r_{k+1}\|_2 \neq 0$  **then**
- 14:      $r_{k+1} *= 1 / \|r_{k+1}\|_2$  normalization
- 15:   **else** exact eigenspace found
- 16:     stop Krylov loop
- 17:   **end if**
- 18: **end for**

**Output:** projection matrix  $B_m := (r_0, \dots, r_{m-1}) \in \mathbb{R}^{N \times m}$  with orthonormal column vectors.

The projected eigenvalue problem (4) with small dimension  $m$  can be solved using a standard eigenvalue solver. Note, that the main costs of the method are related to the total number of time-steps, i.e., the dimension  $m$  of the Krylov space times the number of time steps  $L$  in each Krylov iteration. The orthonormalization of Krylov vectors and the solution of the small eigenvalue problems can be neglected for  $m \ll N$ .

It remains to discuss a stopping criterion, i.e., how to choose the dimension  $m$  of the Krylov space, and a criterion to distinguish converged from non-converged eigenvalues. To this end, we compute for the eigenpairs  $(\omega_{j,m}^2, v_{j,m})$  of (4) the residuals

$$\text{res}_j := \left\| \left( S - \omega_{j,m}^2 M \right) B_m v_{j,m} \right\|_2, \quad j = 1, \dots, m, \quad (17)$$

of the large eigenvalue problem (2). We accept eigenpairs  $(\omega_{j,m}^2, B_m v_{j,m}) \in \mathbb{R} \times \mathbb{R}^N$  for which the residuals are below a given tolerance. A possible stopping criterion would be to increase  $m$  until a fixed number of accepted eigenvalues is found.

Other approaches turned out to be problematic in practice. One idea would be to use the absolute values of the eigenvalues  $\mu_{j,m} := \tilde{\beta}_\alpha(\omega_{j,m})$  of the auxiliary problem (3), which was used to construct the Krylov space. Since we expect convergence towards the eigenvalues  $\mu$  with largest absolute values, we e.g., could accept those eigenpairs  $(\omega_{j,m}^2, B_m v_{j,m})$  for which the ratio  $(|\hat{\mu}|/|\mu_{j,m}|)^m$  with  $|\hat{\mu}| := \min \{ |\mu_{j,m}|, j = 1, \dots, m \}$  is small enough.

Nevertheless, in our numerical experiments sometimes non-converged eigenvalues with large  $|\mu|$  appeared. This might be the case, if eigenvalues within a certain interval are sought. Then, from time to time, a discrete eigenvalue, which will finally converge for  $m \rightarrow N$  from above to an eigenvalue below the region of interest, passes by and leads by accident to a large value  $|\mu|$ . See Fig. 11b for such an example, where even after 50 Krylov steps there are non-converged eigenvalues with large values  $\mu$  and very large residuals. Therefore, we refrained from using the values  $\tilde{\beta}_\alpha(\omega_{j,m})$  in an error indicator.

Another idea would be to use a standard Arnoldi solver for the auxiliary problem (3), i.e., project the matrix  $C$  to the Krylov space, solve the projected eigenvalue problem, use e.g., the residuals of this auxiliary problem as an error indicator, and project the original eigenvalue problem (2) only on the span of those eigenvectors, which were computed and accepted by the auxiliary problem. We refrained from this approach since the residuals of the auxiliary problem are somewhat artificial. The eigenvectors are approximations of the sought eigenvectors, but the eigenvalues  $\mu$  have no one-to-one relation to the eigenvalues  $\omega^2$  or the original eigenvalue problem.

### 3. Numerical experiments

We apply our algorithm to two different sets of problems. In Section 3.1 we tackle a two-dimensional problem where reference solutions can be computed using a shift-and-invert Arnoldi method. In Section 3.2 we choose a three-dimensional problem with more than  $10^6$  unknowns, where the application of reference solutions would require significant computational effort when a direct inverse is used in the process. All numerical examples were carried out using the high-order finite element library Netgen/NGSolve ([12,13]).

#### 3.1. Small scale (2d) examples

The examples in this subsection are chosen in a way that the resulting systems are small enough to allow to compute reference values by applying a shift-and-invert Arnoldi algorithm. The goal of this subsection is to demonstrate the applicability and functionality of our method as well as to give hints on how to choose the parameters.

##### 3.1.1. Description of the experiments

We choose a two-dimensional domain shaped like a dumbbell, consisting of two circles with radii  $r_l, r_r$ , connected by a square with width  $d$  (cf. Fig. 2). We prescribe homogeneous Neumann boundary conditions and choose the geometry parameters

$$r_l = 1.5, \quad r_r = 0.15, \quad d = 0.03. \quad (18)$$

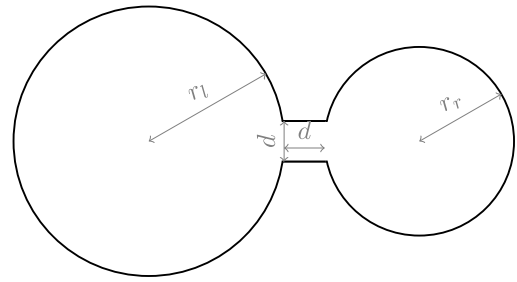


Fig. 2. Sketch of the two-dimensional geometry used for the small scale examples from Section 3.1.

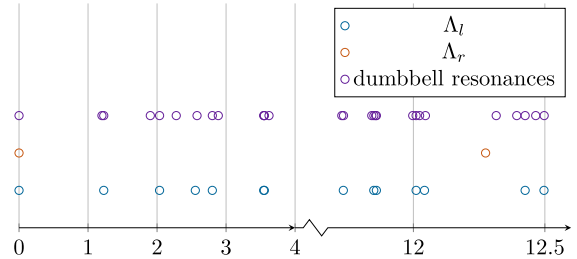


Fig. 3. Exact resonances of (2) on the circles with radii  $r_l = 1.5, r_r = 0.15$  and the whole dumbbell domain (cf. Fig. 2 and (18), (19)).

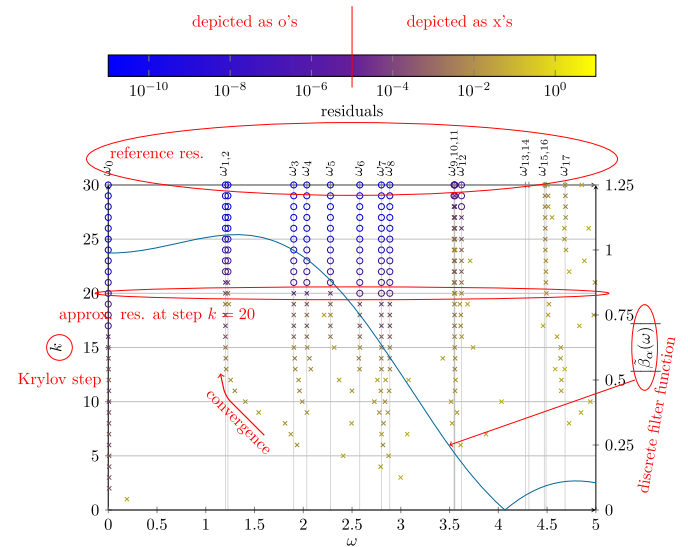


Fig. 4. Numerical results for the computation of resonances of (2) for the dumbbell domain and discrete filter function for  $T = 300\tau, \omega_- = 0, \omega_+ = 3$ . Note that the multiple indices of  $\omega$  separated by commas refer to resonances close to each other and are not resonances of higher multiplicity.

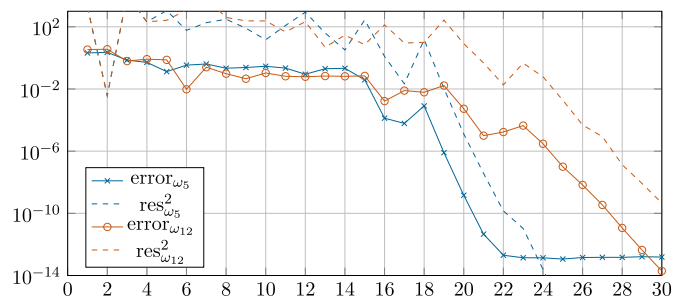


Fig. 5. Errors and residuals for selected resonances of Fig. 4.

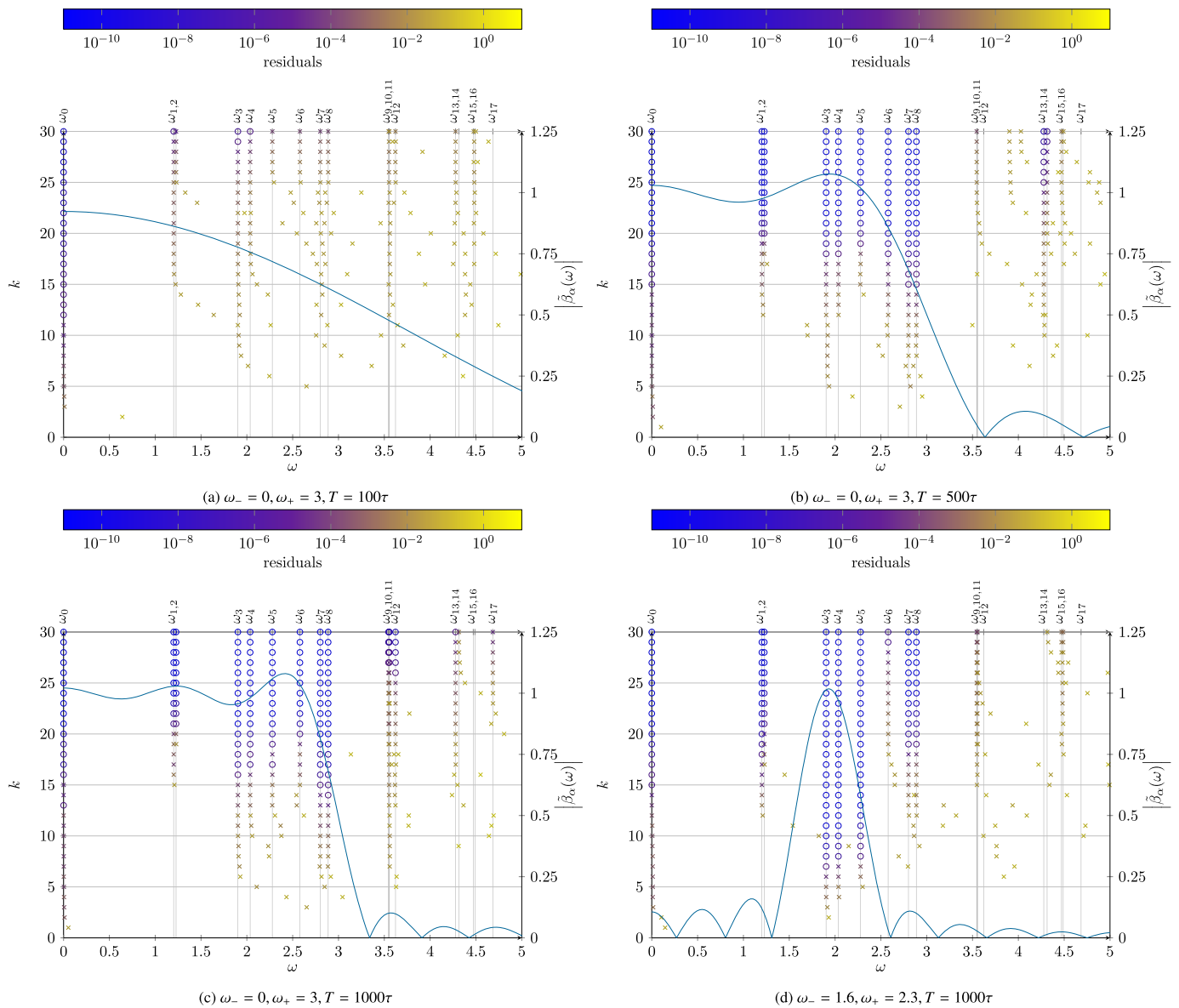


Fig. 6. Convergence of the smallest eigenpairs for different weight function parameters  $\omega_-, \omega_+, T$  (see Fig. 4 for a detailed explanation of the illustration).

For the spacial discretization we employ the mass lumping technique described in Remark 2.1. We choose a mesh-size of  $h = d = 0.03$  (cf. Fig. 7) and second order finite elements. We expect the corresponding discrete resonances  $\omega_j$  to be perturbations of the union of the three sets

$$\Lambda_l = \left\{ \frac{\lambda_{j,n}}{r_l}, j, n \in \mathbb{N}_0 \right\}, \quad \Lambda_r = \left\{ \frac{\lambda_{j,n}}{r_r}, j, n \in \mathbb{N}_0 \right\}, \quad (19)$$

$$\Lambda_d = \left\{ \frac{\pi \sqrt{n^2 + j^2}}{d}, j, n \in \mathbb{N}_0 \right\},$$

where  $\lambda_{j,n}$  are the roots of the derivatives  $J'_n$  of the cylindrical Bessel functions  $J_n$  of the first kind with order  $n \in \mathbb{N}_0$  (cf. [14, (10.2.2)]). Fig. 3 shows the resonances  $\omega$  of the discrete problem (2) together with the sets  $\Lambda_l$  and  $\Lambda_r$ . The set  $\Lambda_d \setminus \{0\}$  is omitted since it consists of high frequencies, which are irrelevant for our computations. In our following experiments we look for the discrete resonances which correspond to perturbations of the smallest non-trivial elements of the sets  $\Lambda_l, \Lambda_r$ .

### 3.1.2. First results

A stable time step  $\tau \approx 0.0056$  is determined using a power iteration and the CFL condition of the time-stepping (11). We choose the weight function  $\alpha$  given in (10) with  $\omega_- = 0, \omega_+ = 3$  and end-time  $T = 300\tau \approx 1.68$ . Fig. 4 shows the resulting approximations to the discrete resonances in each Krylov step  $k$ , as well as the residuals (see (17)), reference values and the discrete filter function  $\tilde{\beta}_\alpha$ . Since we will use this type of illustration throughout the remainder of the paper we use Fig. 4 to explain all the features of this illustration in more detail.

The horizontal axis corresponds to the values of the approximated resonances  $\omega$ . The vertical axis on the one hand, corresponds to the Krylov step  $k$  and on the other hand to the absolute value of the discrete filter function  $\tilde{\beta}_\alpha(\omega)$ . The reference resonances are marked by the vertical grid, thus we observe convergence of the computed values towards these horizontal lines going upward (i.e., increasing the Krylov step). The colors denote the value of the residuals (color bar above the plot), where we use the ring symbol for residuals smaller than  $10^{-5}$  (i.e., we consider the resonance converged) and the cross symbol for residuals larger than  $10^{-5}$ . Fig. 5 confirms that our algorithm converges for

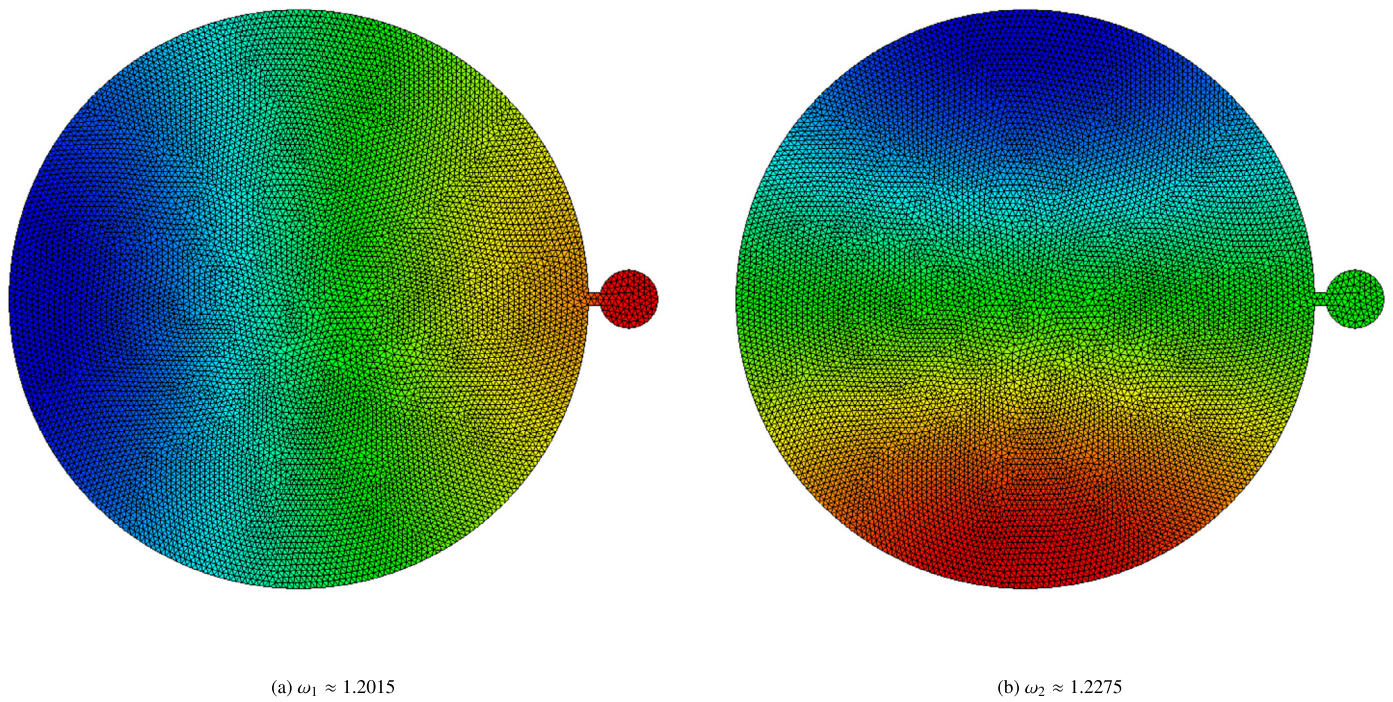


Fig. 7. Eigenfunctions corresponding to the resonances  $\omega_1, \omega_2$  of the 2d example problem, where the green coloring is zero and red/blue marks positive/negative values.

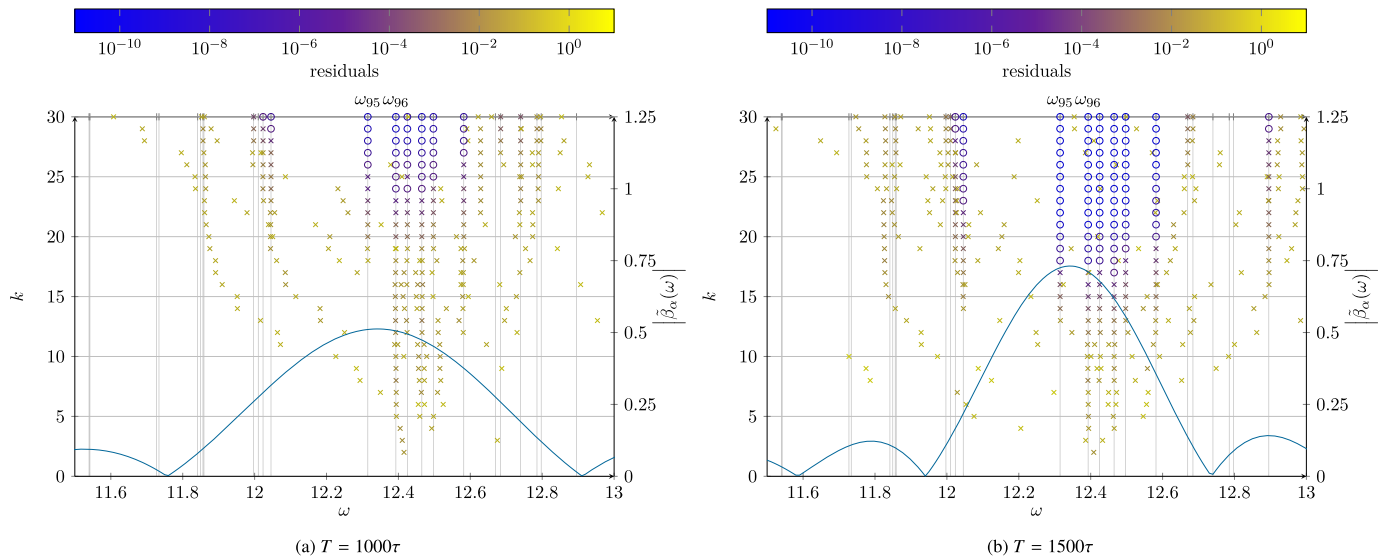


Fig. 8. Convergence of higher frequency eigenpairs for  $\omega_- = 12.2, \omega_+ = 12.5$  and different end times  $T$  (see Fig. 4 for a detailed explanation of the illustration).

two selected resonances and that the squares of the residuals behave asymptotically as the errors.

As to be expected, in general, resonances where the value of the filter function is larger are approximated after fewer Krylov steps: Resonances in the range of 0 to 3 are approximated well at around 20 Krylov steps. The resonances around 3.5 are approximated later. The resonances at around 4.5 still seem to be approximated (but are by far not converged yet). The resonances close to the root of the filter function around 4.3 are not approximated at all using this filter and 30 Krylov steps.

We also note that the convergence does not correspond exactly to the absolute value of the filter function. This is to be expected since the speed of convergence of one resonance also depends on its ratio to neighboring ones, i.e., the convergence of resonances with close absolute values of  $\hat{\beta}_\alpha$  is worse than the convergence of resonances with more isolated absolute values of  $\hat{\beta}_\alpha$ .

### 3.1.3. Different filters

We study the effect of choosing different parameters  $\omega_-, \omega_+$ , and  $T$  on the discrete filter and the approximation of eigenvalues. Fig. 6 shows the approximation of the same problem as before. In Figs. 6a–6c the same interval  $[\omega_-, \omega_+] = [0, 3]$  as before is combined with different numbers of time steps. In Fig. 6d the interval is changed. Due to the few time steps  $T = 100\tau$ , in Fig. 6a the filter function does not have a good contrast for the wanted low frequencies and the unwanted higher frequencies. Thus convergence (except for  $\omega = 0$ ) is only achieved after 30 Krylov steps. For more time steps (cf. Figs. 6b, 6c the contrast is better. This leads to convergence for eight resonances after 20 Krylov steps. However the experiment also shows, that using more time steps does not necessarily lead to better results: The computational effort from Fig. 6b to 6c is doubled, due to the twice as many time steps. This would only pay off if the number of Krylov steps needed to compute the sought resonances is

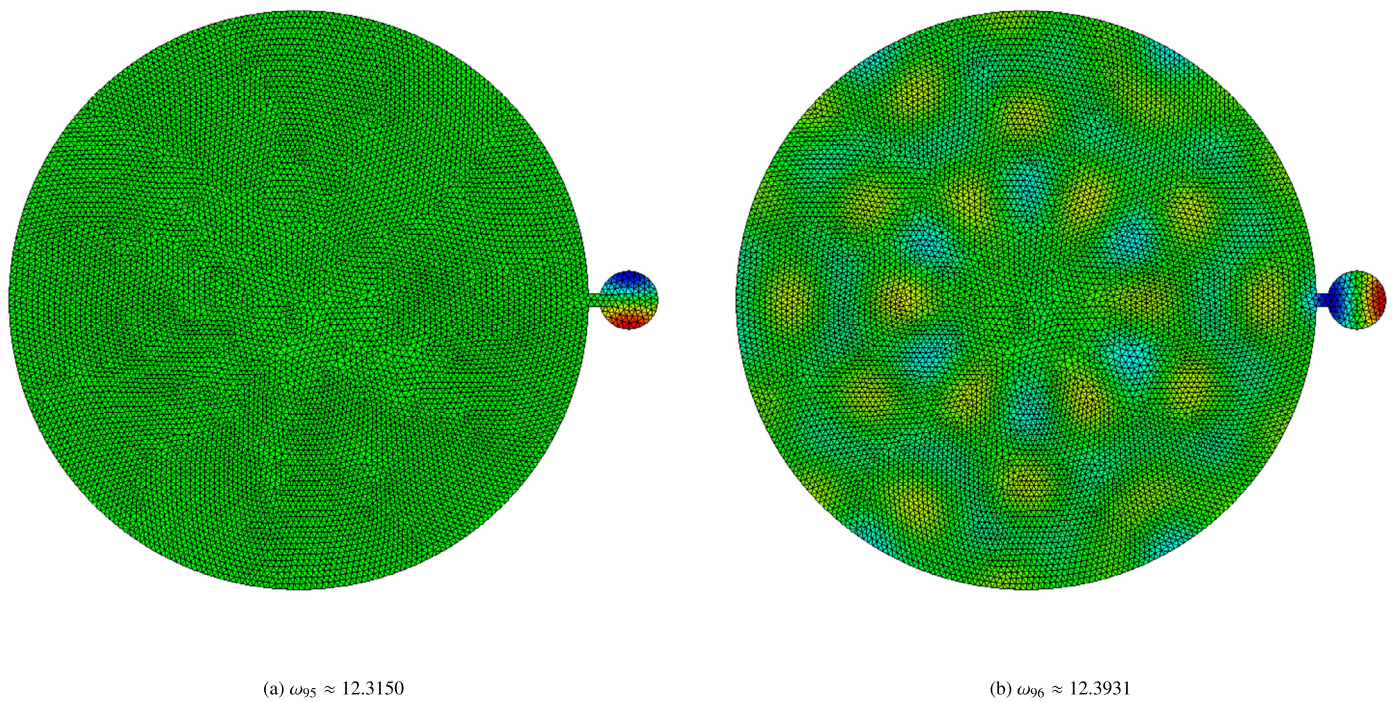


Fig. 9. Eigenfunctions corresponding to the resonances  $\omega_{95}, \omega_{96}$  of the 2d example problem.

more than halved. However using 1000 time steps enables us to choose a narrow range of resonances (i.e.,  $\omega_- = 1.6, \omega_+ = 2.3$ , in Fig. 6d). In this case the resonance  $\omega_3$  is approximated well after 7 Krylov steps (7000 time steps in total) opposed to 18 Krylov steps (9000 time steps in total) in Fig. 6b. To complete the example Fig. 7 shows the first two non-trivial eigenfunctions generated by the large ball. They are perturbations of the eigenfunctions the first non-trivial eigenvalue of multiplicity two of the closed ball. (cf. Section 3.1.1).

To test the filtering of larger resonances we look for the resonance corresponding to the smallest non-trivial element of  $\Lambda_r$  (cf. Fig. 3). To this end we choose  $\omega_- = 12.2, \omega_+ = 12.5$ . The results in Fig. 8 show that the filtering of higher frequency resonances works as expected. However due to the fact that the spectrum of our problem gets denser in higher frequency ranges we have to choose a narrower peak of the filter and thus also more time steps.

Fig. 8b also underlines the necessity to sort out the results based on the residuum, since even in the last Krylov step, there are non-converged resonances in between the converged ones. These resonances will eventually converge with increasing Krylov space dimension to eigenvalues below the region of interest. Finally, Fig. 9 shows that the resonances around 12.3 are in fact perturbations of the resonances of the closed small circle with multiplicity two.

### 3.2. Large scale examples

In this section we apply our method to large examples ( $> 10^6$  unknowns) where methods which require matrix inversion are no longer feasible on desktop computers, due to long factorization times and/or memory requirements. We demonstrate that our method is applicable on off-the-shelf computers.

Our goal is to simulate a hunting horn in a closed room and to find the resonances corresponding to the lowest notes of the horn (i.e., the first few notes of the harmonic series). The horn (cf. Fig. 10) consists of a coiled tube with inner radius  $r$  and thickness  $d$ . The radius of the coil is  $R_c$  and the winding number is 1.35 and coil spacing  $H$ . The mouthpiece is modeled by Dirichlet boundary conditions on one end of the tube. The bell is modeled by the rotation of a spline along the axis with parameters  $R_b, L_b$ .

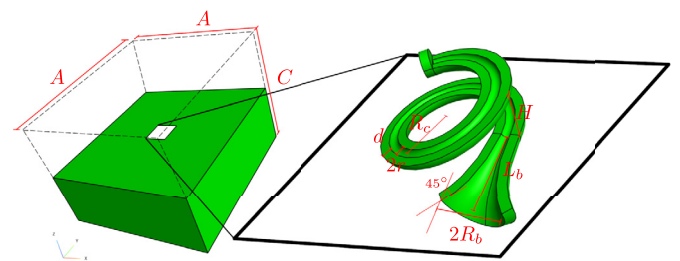


Fig. 10. Cuts through the geometry of the horn and surroundings.

The horn is enclosed in a cuboidal box with dimensions  $A, B$ , and  $C$ . In all our experiments we fixed the parameters

$$\begin{aligned} r &= 0.0123, & d &= 0.0246, & R_c &= 0.15, \\ R_b &= 0.08, & L_b &= 0.2, & H &= 4r + 2d \end{aligned}$$

of the horn and  $A = B = 3, C = 2.4$  for the bounding box. To discretize the problem we use a mesh with mesh-size  $h = d = 0.0246$ . We use a space of first order, mass lumped finite elements. This results in a problem of size  $N \approx 1.245 \cdot 10^6$ . Again, a power iteration determines the time step  $\tau \approx 0.000346$  to be stable.

Similar to the small scale experiments we expect the resonances of the resulting problem to be either perturbations of the resonances of the box (their discrete approximations will be denoted by  $\omega_l^{\text{room}}$  sorted in ascending order in the following) given by

$$\lambda_{i,j,k} = \pi \sqrt{i^2/A + j^2/B + k^2/C}, \quad i, j, k \in \mathbb{N}_0,$$

or perturbations of the resonances of the (closed) inner of the horn (discrete approximations are denoted by  $\omega_l^{\text{horn}}$ ). We look for the latter, since they correspond to the musical notes which can be played on the horn.

The example is chosen as a challenging task for eigenvalue solvers, since the sought resonance frequencies of the horn lie in a region where the background resonances of the box are already quite dense. Note however, that for the efficient computation of such resonances absorbing boundaries/layers like perfectly matched layers (PMLs) or infinite elements may be used.



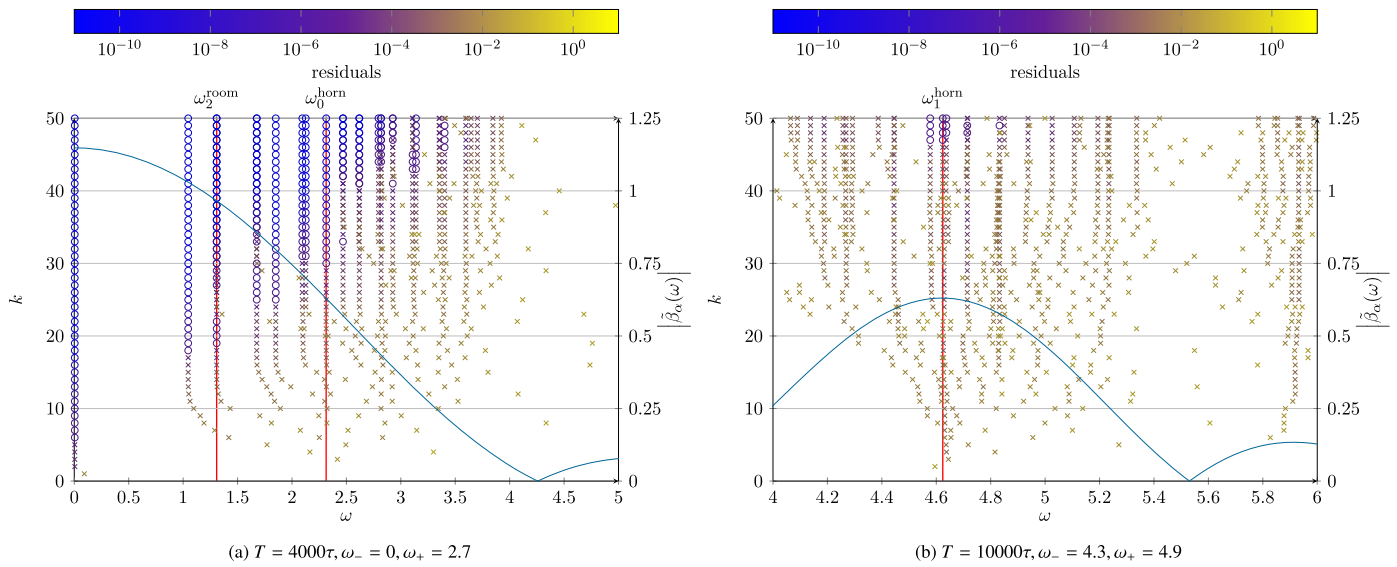


Fig. 11. Computed resonances of the horn domain for different filter parameters  $T, \omega_-, \omega_+$ , where we look for the base harmonics of the horn  $\omega_0^{\text{horn}}, \omega_1^{\text{horn}}$ .

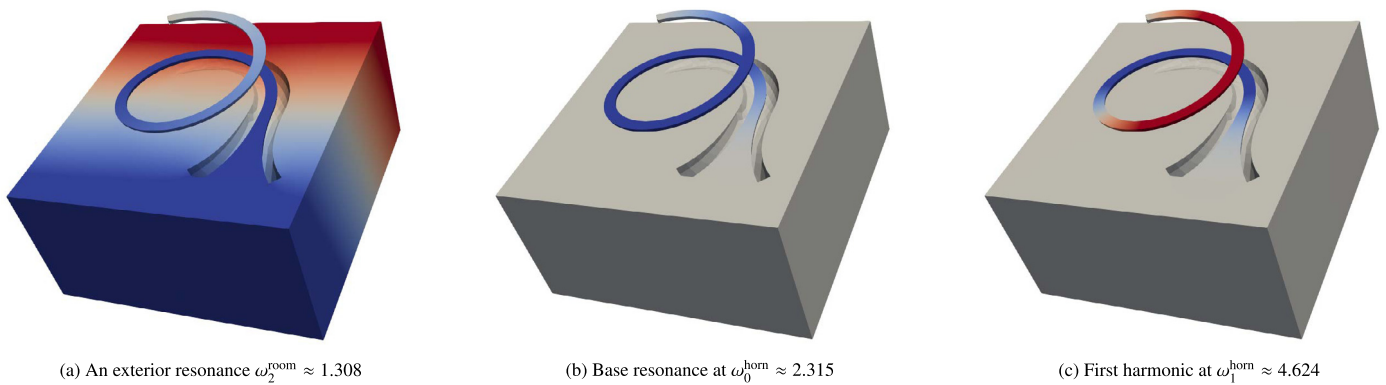


Fig. 12. Eigenfunctions of the horn problem corresponding to the frequencies  $\omega_2^{\text{room}}, \omega_0^{\text{horn}}, \omega_1^{\text{horn}}$  from Fig. 11. The colors blue and red correspond to higher/lower values while gray corresponds to zero. Note that not the whole domain, but merely a cutout (cf., Fig. 10) is shown.

As before we use different sets of parameters for  $T, \omega_-, \omega_+$  to search in different frequency ranges for eigenfrequencies. Using  $T = 4000\tau, \omega_- = 0, \omega_+ = 2.7$  we find the base mode of the horn at  $\omega_0^{\text{horn}} \approx 2.315$  (cf. Figs. 11a, 12b). Thus we look for the second harmonic at approximately  $2\omega_0^{\text{horn}}$ . Fig. 11b shows the result for  $T = 10000, \omega_- = 4.3, \omega_+ = 4.9$ . The spectrum is already very dense in this region, still our algorithm manages to converge for a few resonances. Indeed we find the second harmonic of our horn at  $\omega_1^{\text{horn}} \approx 4.624$  (cf. Fig. 12c).

**Remark 3.1.** If we assume the dimensions of our horn to be given in meters and a speed of sound of  $343\text{m/s}$  we obtain that the first two notes which can be played have frequencies of

$$343 \frac{\omega_0^{\text{horn}}}{2\pi} \approx 126.38 \text{ Hz}, \quad 343 \frac{\omega_1^{\text{horn}}}{2\pi} \approx 252.43 \text{ Hz}.$$

In musical notation this corresponds to notes a little higher than a great and small B respectively. This is a reasonable result given the fact that our horn has a total length of  $1.479\text{m}$ , which is in a similar range as a trumpet tuned in B flat (usually a length of approximately  $1.485\text{m}$ ).

#### 4. Generalizations and extensions

In Section 2 and the corresponding numerical experiments in Section 3 we focused on applying our ideas to an example composed of the following components:

- discrete problem:** The generalized matrix eigenvalue problem (2) stems from a finite element discretization of a Laplacian eigenvalue problem using mass lumping.
- time-stepping:** The discrete filter function defined in (15) is based on the Verlet time-stepping of the time-domain problem (5) corresponding to (2) and a suitable
- weight function:** The weight function is based on an approximation of the inverse Fourier transform of an indicator function (cf. Equations (9) and (10)).
- eigenvalue solver:** To approximate the eigenvectors of the auxiliary eigenvalue problem (3) we construct a basis of the Krylov space of the operator  $C$  by iterative application and orthonormalization.

In fact, all of these components can be replaced by suitable alternatives. In the following we discuss such extensions of the method, as well as the limits of the current approach.

##### 4.1. Discrete problem

The main requirement for the method to work is the fact that the matrix eigenvalue problem (2) corresponds to a stable time-domain problem of the form (5). This is certainly the case if the matrices  $M$  and  $S$  are symmetric and positive definite and positive semi-definite respectively. This is fulfilled for suitable discretizations of wave-type equations where  $S$  is the discrete representation of any elliptic, second

order differential operator. Examples include the curl curl-operator (for linear Maxwell equations) or the elastic operator.

The main requirement for the method to be efficient is the fact that the according time-domain problem can be efficiently approximated. For explicit time-stepping methods (cf. Section 4.2) this is the case if the inverse of the mass matrix  $M$  can be applied efficiently. Apart from mass lumping techniques this is also the case for finite difference methods, discontinuous Galerkin approaches (see e.g., [15]) or cell methods [16, 17].

Going further one could even drop the assumption that the problem is self-adjoint (Hermitian) if there is a stable time-domain counterpart. One example could be resonance problems in open systems, which are discretized using perfectly matched layers (cf. [18]). However, the choice of the weight function  $\alpha$  is not a priori clear if complex resonances exist.

#### 4.2. Time-stepping

Following Remark 2.6 the Verlet time-stepping (11) can be easily replaced by any other explicit time-stepping method, if the eigenvectors of (2) are still eigenvectors of (3). For first order formulations one could e.g., use leap frog time-stepping or high-order variants thereof to construct a similar method. Time-stepping schemes where the correspondence of eigenvectors as described above is not clear any more include local time-stepping [19] or locally implicit methods [20]. These methods have been proven to significantly reduce the CFL condition on the time-step size which is a major limiting factor in computational efficiency. However, they do not fit directly into the concept presented in this paper.

#### 4.3. Weight function

The weight function chosen in (10) is based on the inverse Fourier transform of a characteristic function. As an alternative to the characteristic function one could choose more localized and/or smoother functions (e.g., a triangle impulse or a Gaussian peak) to construct the weight function  $\alpha$ . In particular, the weight function  $\alpha$  could be adapted, such that in regions where an essential spectrum or an eigenvalue with very high multiplicity is expected, the absolute values of the corresponding discrete filter  $\tilde{\beta}_\alpha$  are small. An example where this would be useful is e.g., the infinite dimensional kernel of the curl curl-operator.

#### 4.4. Eigenvalue solver

To resolve issues with multiple eigenvalues a block version of the Krylov space approach can be used (i.e., starting with an orthonormalized set of random vectors and also orthonormalizing the resulting vectors in each step). Alternatively instead of considering a growing Krylov space in each step one could fix the dimension which leads to a method corresponding to a FEAST [2] algorithm with a time-domain filter.

### 5. Conclusion

We have presented an eigenvalue solver for large scale eigenvalue problems originating from finite element discretizations of Laplacian eigenvalue problems. Existing explicit time-domain solvers are combined with a weight function in order to focus a Krylov method to eigenvalues which might be non-extremal and/or clustered. It is straightforward to show that the method converges to those eigenvalues, which are mapped by the discrete filter  $\tilde{\beta}_\alpha$  defined in (15) to eigenvalues with

largest absolute value of the auxiliary eigenvalue problem. A complete error analysis would need to control the absolute values of the filtered sought and unsought eigenvalues, which lies out of the scope of this paper.

Note, that the discrete filter depends on the weight function  $\alpha$ , the number and size of time-steps, and on the time-stepping scheme. The filter function can be computed in a preprocessing step with negligible computational costs. Hence, an optimization of the method parameters for a given problem is easy to carry out experimentally.

### Acknowledgements

The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

### Data availability

No data was used for the research described in the article.

### References

- [1] Y. Saad, Numerical Methods for Large Eigenvalue Problems, Classics in Applied Mathematics, vol. 66, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2011, revised edition of the 1992 original [1177405].
- [2] E. Polizzi, Density-matrix-based algorithm for solving eigenvalue problems, Phys. Rev. B 79 (2009) 115112.
- [3] J. Gopalakrishnan, L. Grubišić, J. Owall, Spectral discretization errors in filtered subspace iteration, Math. Comput. 89 (321) (2020) 203–228.
- [4] T. Sakurai, H. Sugiura, A projection method for generalized eigenvalue problems using numerical integration, in: Proceedings of the 6th Japan-China Joint Seminar on Numerical Mathematics, Tsukuba, 2002, vol. 159, 2003, pp. 119–128.
- [5] W.-J. Beyn, An integral method for solving nonlinear eigenvalue problems, Linear Algebra Appl. 436 (10) (2012) 3839–3863.
- [6] A.V. Knyazev, Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method, in: Copper Mountain Conference (2000), vol. 23, 2001, pp. 517–541.
- [7] M.J. Grote, F. Nataf, J.H. Tang, P.-H. Tournier, Parallel controllability methods for the Helmholtz equation, Comput. Methods Appl. Mech. Eng. 362 (2020) 112846.
- [8] D. Appelö, F. Garcia, O. Runborg, WaveHoltz: iterative solution of the Helmholtz equation via the wave equation, SIAM J. Sci. Comput. 42 (4) (2020) A1950–A1983.
- [9] C.C. Stolk, A time-domain preconditioner for the Helmholtz equation, SIAM J. Sci. Comput. 43 (5) (2021) A3469–A3502.
- [10] R. Li, Y. Xi, E. Vecharynski, C. Yang, Y. Saad, A thick-restart Lanczos algorithm with polynomial filtering for Hermitian eigenvalue problems, SIAM J. Sci. Comput. 38 (4) (2016) A2512–A2534, <https://doi.org/10.1137/15M1054493>.
- [11] G. Cohen, P. Joly, J.E. Roberts, N. Tordjman, Higher order triangular finite elements with mass lumping for the wave equation, SIAM J. Numer. Anal. 38 (6) (2001) 2047–2078.
- [12] J. Schöberl, Netgen - an advancing front 2d/3d-mesh generator based on abstract rules, Comput. Vis. Sci. 1 (1997) 41–52.
- [13] J. Schöberl, C++11 implementation of finite elements in ngsolve, Preprint 30/2014, Institute for Analysis and Scientific Computing, TU Wien, 2014.
- [14] NIST Digital Library of Mathematical Functions, <https://dlmf.nist.gov/>, Release 1.1.12 of 2023-12-15, f.W.J. Olver, A.B. Olde Daalhuis, D.W. Lozier, B.I. Schneider, R.F. Boisvert, C.W. Clark, B.R. Miller, B.V. Saunders, H.S. Cohl, and M.A. McClain, eds.
- [15] J.S. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods, Texts in Applied Mathematics, vol. 54, Springer, New York, New York, NY, 2008.
- [16] M. Wess, B. Kapidani, L. Codecasa, J. Schöberl, Mass lumping the dual cell method to arbitrary polynomial degree for acoustic and electromagnetic waves, J. Comput. Phys. 513 (2024) 113196, <https://doi.org/10.1016/j.jcp.2024.113196>.
- [17] B. Kapidani, L. Codecasa, J. Schöberl, An arbitrary-order cell method with block-diagonal mass-matrices for the time-dependent 2d Maxwell equations, J. Comput. Phys. 433 (2021) 110184.
- [18] J.-P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. 114 (2) (1994) 185–200.
- [19] J. Diaz, M.J. Grote, Energy conserving explicit local time stepping for second-order wave equations, SIAM J. Sci. Comput. 31 (3) (2009) 1985–2014.
- [20] S. Piperno, Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems, M2AN Math. Model. Numer. Anal. 40 (5) (2006) 815–841.