THE WAVEGUIDE EIGENVALUE PROBLEM AND THE TENSOR INFINITE ARNOLDI METHOD

ELIAS JARLEBRING, GIAMPAOLO MELE, OLOF RUNBORG*

Abstract. We present a new computational approach for a class of large-scale nonlinear eigenvalue problems (NEPs) that are nonlinear in the eigenvalue. The contribution of this paper is two-fold. We derive a new iterative algorithm for NEPs, the tensor infinite Arnoldi method (TIAR), which is applicable to a general class of NEPs, and we show how to specialize the algorithm to a specific NEP: the waveguide eigenvalue problem. The waveguide eigenvalue problem arises from a finite-element discretization of a partial differential equation (PDE) used in the study waves propagating in a periodic medium. The algorithm is successfully applied to accurately solve benchmark problems as well as complicated waveguides. We study the complexity of the specialized algorithm with respect to the number of iterations m and the size of the problem n, both from a theoretical perspective and in practice. For the waveguide eigenvalue problem, we establish that the computationally dominating part of the algorithm has complexity $\mathcal{O}(nm^2 + \sqrt{n}m^3)$. Hence, the asymptotic complexity of TIAR applied to the waveguide eigenvalue problem, for $n \to \infty$, is the same as for Arnoldi's method for standard eigenvalue problems.

1. Introduction. Consider the propagation of waves in a periodic medium, which are governed by the Helmholtz equation

$$\Delta v(x,z) + \omega^2 \eta(x,z)^2 v(x,z) = 0, \qquad (x,z) \in \mathbb{R}^2, \tag{1.1}$$

where $\eta \in L^\infty(\mathbb{R}^2)$ is called the index of refraction and ω the temporal frequency. When (1.1) models an electromagnetic wave, the solution v typically represents the y-component of the electric or the magnetic field. The (spatially dependent) wavenumber is $\kappa(x,z) := \omega\eta(x,z)$ and we assume that the material is periodic in the z-direction and without loss of generality the period is assumed to be 1, i.e., $\eta(x,z+1) = \eta(x,z)$. The index of refraction is assumed to be constant for sufficiently large $|x|$, such that $\kappa(x,z) = \kappa_-$ when $x < x_-$, $\kappa(x,z) = \kappa_+$ when $x > x_+$. In this paper we assume the wavenumber to be piecewise constant. Figure 1.1 shows an example of the setup.

Bloch solutions to (1.1) are those solutions that can be factorized as a product of a z-periodic function and $e^{\gamma z}$, i.e.,

$$v(x,z) = \hat{v}(x,z)e^{\gamma z}, \qquad \hat{v}(x,z+1) = \hat{v}(x,z). \tag{1.2}$$

The constant $\gamma \in \mathbb{C}$ is called the Floquet multiplier and without loss of generality, it is assumed that $\operatorname{Im}\gamma \in (-2\pi, 0]$. We interpret (1.1) in a weak sense. We are only interested in Bloch solutions that decay in magnitude as $|x| \to \infty$ and we require that $\hat{v}$, restricted to $S := \mathbb{R} \times (0,1)$, belongs to the Sobolev space $H^1(S)$. Moreover, we assume that any Bloch solution has a representative in $C^1(S)$. These solutions are in general not in $C^2(S)$ since κ is discontinuous.

In this contex, Bloch solutions are also called guided modes of (1.1). If γ is purely imaginary, the mode is called *propagating*; if $|\operatorname{Re}\gamma|$ is small it is called *leaky*. Both mode types are of great interest in various settings [11, 26, 31, 28, 2]. We present a procedure to compute leaky modes, with $\operatorname{Re}\gamma < 0$ and $\operatorname{Im}\gamma \in (-2\pi, 0)$. This specific setup has been studied, e.g., in [32].

To compute the guided modes one can either fix ω and find γ, or, conversely, fix γ and find ω. Both formulations lead to a PDE eigenvalue problem set on the unbounded

*Dept. Mathematics, KTH Royal Institute of Technology, SeRC swedish e-science research center, Lindstedtsvägen 25, Stockholm, Sweden, email: {eliasj,gmele,olofr}@kth.se

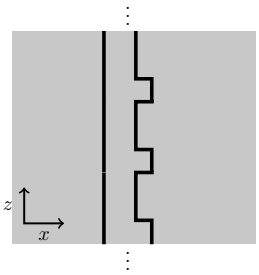


Figure 1.1: Illustration of a waveguide defined on $\mathbb{R}^2$. The wavenumber $\kappa(x,z)$ is constant in the three regions separated by the thick lines.

domain S. When γ is held fix, the eigenvalue problem is linear and if ω is held fix, it is nonlinear (quadratic) in γ. In this paper we fix ω, and the substitution of (1.2) into (1.1) leads to the following problem. Find $(\gamma, \hat{v}) \in \mathbb{C} \times H^1(S)$ such that

$$\Delta\hat{v}(x,z) + 2\gamma\hat{v}_z(x,z) + (\gamma^2 + \kappa(x,z)^2)\hat{v}(x,z) = 0, \quad (x,z) \in S, \tag{1.3a}$$

$$\hat{v}(x,0) = \hat{v}(x,1), \quad x \in \mathbb{R}, \tag{1.3b}$$

$$\hat{v}_z(x,0) = \hat{v}_z(x,1), \ x \in \mathbb{R}. \tag{1.3c}$$

The problem (1.3), which in this paper is referred to as the waveguide eigenvalue problem, is defined on an unbounded domain. We use a well-known technique to reduce the problem on a unbounded domain to a problem on a bounded domain. We impose artificial (absorbing) boundary conditions, in particular so-called Dirichlet-to-Neumann (DtN) maps. See [14, 3] for literature on artificial boundary conditions.

The DtN-reformulation and a finite-element discretization, with rectangular elements generated by a uniform grid with n_x and n_z grid points in x and z-direction correspondingly, is presented in Section 2. A similar DtN-discretization has been applied to the waveguide eigenvalue problem in the literature [32]. In relation to [32], we need further equivalence results for the DtN-operator and use a different type of discretization, which allows easier integration with our new iterative method. Due to the fact that the DtN-maps depend on γ, the discretization leads to a nonlinear eigenvalue problem (NEP) of the following type. Find $(\gamma, w) \in \mathbb{C} \times \mathbb{C}^n \backslash \{0\}$ such that

$$M(\gamma)w = 0, \tag{1.4}$$

where

$$M(\gamma) := \begin{bmatrix} Q(\gamma) & C_1(\gamma) \\ C_2^T & P(\gamma) \end{bmatrix} \in \mathbb{C}^{n \times n}, \tag{1.5}$$

and $n = n_x n_z + 2n_z$. The matrices $Q(\gamma) \in \mathbb{C}^{n_x n_z \times n_x n_z}$ and $C_1(\gamma) \in \mathbb{C}^{n_x n_z \times 2n_z}$ are a quadratic polynomials in γ, $Q(\gamma) = A_0 + A_1\gamma + A_2\gamma^2$, $C_1(\gamma) = C_{1,0} + C_{1,1}\gamma + C_{1,2}\gamma^2$, where $A_i, C_{1,i}$ and C_2^T are large and sparse. The matrix $P(\gamma)$ has the structure

$$P(\gamma) = \begin{bmatrix} R\Lambda_-(\gamma)R^{-1} & 0 \\ 0 & R\Lambda_+(\gamma)R^{-1} \end{bmatrix} \in \mathbb{C}^{2n_z \times 2n_z} \tag{1.6}$$

and $\Lambda_\pm(\gamma) \in \mathbb{C}^{n_z \times n_z}$ are diagonal matrices containing nonlinear functions of γ involving square roots of polynomials. The matrix-vector product corresponding to R and R^{-1} can be computed with the Fast Fourier Transform (FFT).

We have two main contributions in this paper:

- a new algorithm, the tensor infinite Arnoldi method (TIAR), for a general class of NEPs (1.4), which is based on a tensor representation of the basis of the infinite Arnoldi method (IAR) [18];
- a discretization of the waveguide eigenvalue problem and an adaption of TIAR, such that the problem can be efficiently solved.

The contributions are tightly connected since the selection of discretization is done with the objective to be able to use TIAR, and TIAR is further improved by exploitation of the structures of the discretization.

The general NEP (1.4) has received considerable attention in the literature in various generality settings. We list those algorithm that are related to our method. See the review papers [24, 37] and the problem collection [4], for further literature.

The structure of the basis matrix arising in Arnoldi's method in the context of NEPs has been exploited in various settings. Although the approaches are different, they do have in common that they exploit a redundancy in the Krylov subspace. To our knowledge, the first approach was the SOAR-method [1] which is derived from a structure arising in Arnoldi's method applied to a particular companion matrix for quadratic eigenvalue problems. For more general polynomial eigenvalue problems, an approach was presented at a conference [30]. The approach [21] also contains a structure exploitation designed for polynomial eigenvalue problems (expressed in a Chebyshev basis). It is particularly suitable to use in a two stage-approach, which is done in [7], where the eigenvalues of interest lie in a predefined interval and (a non-polynomial) M can first be approximated with interpolation on a Chebyshev grid and subsequently the polynomial eigenvalue problem can be solved with [21]. The algorithm in [39] also exploits a compact representation. It is mainly developed for moment-matching in model reduction of time-delay systems, where the main goal is to compute a subspace (of $\mathbb{C}^n$) with appropriate approximation properties. The algorithm in [36], which has been developed in parallel independent of our work, also exploits compact representation and is based on a rational Krylov method. We further relate to [36] in Section 5.3. In this paper we prove the existence of a compact representation of the basis in the infinite Arnoldi method, which is a method designed for non-polynomial analytic nonlinear eigenvalue problems. We also show how the compact representation can be exploited. The algorithm improves efficiency both in terms of memory and computation time. Moreover, our compact representation can be naturally combined with the waveguide eigenvalue problem.

Some recent approaches for (1.4) exploit low-rank properties, e.g., $M^{(i)}(0) = V_i Q^T$, where $V_i, Q \in \mathbb{C}^{n\times r}$ for sufficiently large i, and r is small relative to n. See, e.g., [29, 38, 34]. This property is present here if we select $r = n_z = \mathcal{O}(\sqrt{n})$, which is not very small with respect to the size of the problem, making the low-rank methods to not appear favorable for this NEP.

The (non-polynomial) nonlinearities in our approach stem from absorbing boundary conditions. Other absorbing boundary conditions also lead to NEPs. This has been illustrated in specific applications, e.g., in the simulation of optical fibers [19], cavity in accelerator design [22], double-periodic photonic crystals [8, 9] and micro-electromechanical systems [5]. There is to our knowledge no approach that integrates the structure of the discretization of the PDE and the γ-dependent boundary conditions with an Arnoldi method. The adaption of the algorithm to our specific PDE is presented in Section 4.

The notation is mostly standard. A matrix consisting of elements $a_{i,j}$ is denoted

$$[a_{i,j}]_{i,j=1}^{m} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,m} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,m} \end{bmatrix}.$$

The notation is analogous for vectors and tensors. We use $\underline{Q}$ to denote an extension of Q with one block row of zeros. The size of the block will be clear by the context.

2. Derivation of the NEP.

2.1. DtN reformulation. Our computational approach is based on the concept of artificial boundary conditions. The technique of artificial boundary conditions has been used in various settings, e.g., [32, 20, 15, 11, 12]. The unbounded-domain problem is equivalently rephrased as a bounded-domain problem on $S_0 := [x_-, x_+] \times [0, 1] \subset S$ by introducing certain boundary conditions at $x = x_\pm$. The boundary conditions are stated in terms of so-called Dirichlet-to-Neumann (DtN) maps which relate the (normal) derivative of the solution at the boundary with the function value at the boundary. The artificial boundary conditions and the discretization are selected such that we can integrate the structure of the discretization, with algorithm presented in Section 4.

We first derive some results necessary for our setting. The DtN formulation of the eigenvalue problem (1.3) is given as follows. Find γ and $u \in H^1(S_0)$ such that

$$\Delta u(x,z) + 2\gamma u_z(x,z) + (\gamma^2 + \kappa(x,z)^2)u(x,z) = 0, \qquad (x,z) \in S_0, \tag{2.1a}$$
$$u(x,0) = u(x,1), \qquad x \in (x_-, x_+), \tag{2.1b}$$
$$u_z(x,0) = u_z(x,1), \qquad x \in (x_-, x_+), \tag{2.1c}$$
$$\mathcal{T}_{-,\gamma}[u(x_-,\cdot)] = -u_x(x_-,\cdot), \tag{2.1d}$$
$$\mathcal{T}_{+,\gamma}[u(x_+,\cdot)] = u_x(x_+,\cdot), \tag{2.1e}$$

where $\mathcal{T}_{\pm,\gamma} : H^1([0,1]) \mapsto L^2([0,1])$ are the DtN maps, defined by

$$\mathcal{T}_{\pm,\gamma}[g](z) := \sum_{k\in\mathbb{Z}} s_{\pm,k}(\gamma) g_k e^{2\pi i k z}, \tag{2.2}$$

where $[g_k]_{k\in\mathbb{Z}}$ is the Fourier expansion of g, i.e., $g(z) := \sum_{k\in\mathbb{Z}} g_k e^{2\pi i k z}$ and

$$\beta_{\pm,k}(\gamma) := (\gamma + 2i\pi k)^2 + \kappa_\pm^2 = ((\gamma + 2i\pi k) + i\kappa_\pm)((\gamma + 2i\pi k) - i\kappa_\pm), \tag{2.3}$$
$$s_{\pm,k}(\gamma) := \operatorname{sign}(\operatorname{Im}(\beta_{\pm,k}(\gamma)))i\sqrt{\beta_{\pm,k}(\gamma)}. \tag{2.4}$$

In this section we show that, under the assumption that neither the real nor the imaginary part of γ vanishes, the DtN maps are well-defined and the problems (1.3) and (2.1) are equivalent. In order to characterize the DtN maps, we consider the exterior problems, i.e., the problems corresponding to the domains $S_+ = (x_+, \infty) \times (0, 1)$ and $S_- = (-\infty, x_-) \times (0, 1)$. The exterior problems are defined as the two problems corresponding to finding $w \in H^1(S_\pm)$ such that, for a given g,

$$\Delta w + 2\gamma w_z + (\gamma^2 + \kappa_\pm^2)w = 0, \qquad (x,z) \in S_\pm, \tag{2.5a}$$
$$w(x,0) = w(x,1), \tag{2.5b}$$
$$w_z(x,0) = w_z(x,1), \tag{2.5c}$$
$$w(x_\pm, z) = g(z). \tag{2.5d}$$

Remark 2.1 (Regularity). *Note that if we multiply a solution to* (1.3), (2.1) *or* (2.5) *with* $e^{\gamma z}$, *we have a solution to the Helmholtz equation, i.e., it satisfies* (1.1) *in their respective domains, i.e.,* S, S_0 *and* $S_\pm$. *By assumption, solutions to* (1.3), (2.1) *and* (2.5) *are* C^1 *and the traces taken on* $x = x_\pm$ *and its first derivatives are always well-defined and continuous. Moreover, for* $x < x_-$ *and* $x > x_+$, *since* $\kappa(x, z)$ *is constant, the problem can be interpreted in a strong sense and the solutions are in* C^∞.

Our assumption that the solution has regularity C^1 *can be relaxed as follows. If we select* x_- *and* x_+ *such that* κ *is constant over* x_- *and* x_+, *we have by elliptic regularity [10, Section 6.3.1, Theorem 1], that weak* H^1 *solutions of* (1.3), (2.1) *and* (2.5) *are in* H^2_{loc}. *This means that traces taken on* $x = x_\pm$ *of a solution and its derivatives are always well-defined and smooth, without explicitly assuming that the solution is in* C^1.

The following result illustrates that the application of the DtN maps in (2.2) is in a sense equivalent to solving the exterior problems and evaluating the solutions in the normal direction at the boundary $x = x_\pm$. More precisely, the following lemma shows that if $\operatorname{Re}\gamma \neq 0$ and $\operatorname{Im}\gamma \in (-2\pi, 0)$ the problems are well-posed in $H^1(S_\pm)$ and the boundary relations (2.1d) and (2.1e) are satisfied. The proof is available in Appendix A.

LEMMA 2.2 (Characterization of DtN maps). *Suppose* $\operatorname{Re}\gamma \neq 0$, *and* $\operatorname{Im}\gamma \notin 2\pi\mathbb{Z}$ *and* $g \in H^{1/2}([0,1])$. *Then, each of the exterior problems* (2.5) *have a unique solution* $w \in H^1(S_\pm)$. *Moreover, there is a constant* C *independent of* g *such that*

$$||w||_{H^1(S_\pm)} \leq C||g||_{H^{1/2}([0,1])}. \tag{2.6}$$

If it is further assumed that $g \in H^1([0,1])$, *then the DtN maps in* (2.2) *are well-defined and satisfy*

$$\mathcal{T}_{+,\gamma}[w(x_+,\cdot)](z) = w_x(x_+,z), \qquad \mathcal{T}_{-,\gamma}[w(x_-,\cdot)](z) = -w_x(x_-,z). \tag{2.7}$$

This lemma immediately implies the equivalence between (1.3) and (2.1) under the same conditions on γ.

THEOREM 2.3 (Equivalence of (1.3) and (2.1)). *Suppose* $\operatorname{Re}\gamma \neq 0$ *and* $\operatorname{Im}\gamma \notin 2\pi\mathbb{Z}$. *Then* $u \in H^1(S_0)$ *is a solution to* (2.1) *if and only if there exists a solution* $\hat{v} \in H^1(S)$ *to* (1.3) *such that* u *is the restriction of* $\hat{v}$ *to* S_0.

Proof. Suppose $\hat{v}$ is a solution of (1.3) and u is its restriction to S_0. Then u clearly satisfies (2.1a-c). By Remark 2.1 the functions $\hat{v}(x_\pm, z) = u(x_\pm, z)$ are in $C^1([0,1]) \subset H^1([0,1])$. Lemma 2.2 shows that $\hat{v}$, restricted to $S_\pm$, are the unique solutions to the exterior problems (2.5). Hence, $\hat{v}$ is identical to the union of u and the solutions to the exterior problems (2.5). Since $\hat{v} \in C^1(S)$, we have that $\hat{v}_x(x_\pm, z)$ is continuous and $w_x(x_\pm, z) = u_x(x_\pm, z) = \hat{v}_x(x_\pm, z)$. Moreover, due to (2.7), the boundary conditions (2.1d-e) are satisfied.

On the other hand, suppose $u \in H^1(S_0)$ is a weak solution to (2.1). Remark 2.1 again implies that $u \in C^1(S_0)$ and in particular $u(x_\pm, z) \in C^1([0,1]) \subset H^1([0,1])$. We have from Lemma 2.2 that the exterior problems (2.5) have unique solutions w that satisfy (2.7). Let $\hat{v}$ be defined as the union of the u and w. The union $\hat{v}$ has a continuous derivative on the boundary $x = x_\pm$ due to (2.1)d-e and (2.7) and since $u \in H^1(S_0)$ and $w \in H^1(S_\pm)$, then $\hat{v} \in H^1(S)$ and satisfies (1.3) by construction. ∎

Remark 2.4 (Conditions on γ). *Modes with* $\operatorname{Re}\gamma = 0$ *are propagating. For those modes, the well-posedness of the DtN-maps depends on the wave number. See [11] for precise results about well-posedness in the situation* $\operatorname{Re}\gamma = 0$. *In our setting we only consider leaky modes and* $\operatorname{Re}\gamma < 0$. *The situation* $\operatorname{Re}\gamma > 0$ *can be treated analogously.*

2.2. Discretization. We discretize the finite-domain PDE (2.1) with a finite-element approach. The domain $[x_-, x_+] \times [0, 1]$ is partitioned using rectangular elements obtained with a uniform distribution of nodes in the x and z directions. We use n_x grid points in the x-direction and n_z grid points in the z-direction and define $x_i = x_- + ih_x$ and $z_j = jh_z$ where $i = 1, \ldots, n_x$, $j = 1, \ldots, n_z$, $h_x = \frac{x_+ - x_-}{n_x+1}$ and $h_z = \frac{1}{n_z}$. The basis functions are chosen as piecewise bilinear functions that are periodic in the z direction with period 1. In particular, the basis functions that we consider are periodic modification of the standard basis functions. We let $\hat{u}$ denote the vector of containing coefficients for the interior points, $\hat{u} \approx \text{vec}([u(z_j, x_i)]_{j=1,i}^{n_x,n_z})$ and $\hat{u}_+$ and $\hat{u}_-$ the coefficients corresponding to the boundary, i.e., $\hat{u}_\pm^T \approx [u(x_\pm, z_1), \ldots, u(x_0, z_{n_x})]$. By applying the Ritz–Galerkin discretization on the weak formulation, we find that

$$Q(\gamma)\hat{u} + C_1(\gamma) \begin{bmatrix} \hat{u}_- \\ \hat{u}_+ \end{bmatrix} = 0, \tag{2.8}$$

where $Q(\gamma) := A_0 + \gamma A_1 + \gamma^2 A_2$, $C_1(\gamma) := C_{1,0} + \gamma C_{1,1} + \gamma^2 C_{1,2}$. The matrices $(A_i)_{i=0}^2$ and $(C_{1,i})_{i=0}^2$ can be computed in an efficient and explicit way[1] with the procedure outlined in Appendix B.

Two approximations must be done in order to incorporate the boundary conditions. We construct approximations of the right-hand side of (2.1)d-e using the one-sided second-order finite-difference approximation,

$$\begin{bmatrix} -u_x(x_-, z_1) \\ \vdots \\ -u_x(x_-, z_{n_z}) \end{bmatrix} \approx -C_{2,-}^T \hat{u} - d_0 \hat{u}_- \text{ and } \begin{bmatrix} u_x(x_+, z_1) \\ \vdots \\ u_x(x_+, z_{n_z}) \end{bmatrix} \approx -C_{2,+}^T \hat{u} - d_0 \hat{u}_+ \tag{2.9}$$

where $C_{2,-}^T = (d_1, d_2, 0, \ldots, 0) \otimes I_{n_z} \in \mathbb{C}^{n_z \times n_z n_x}$ and $C_{2,+}^T = ((0, \ldots, 0, d_2, d_1) \otimes I_{n_z}) \in \mathbb{C}^{n_z \times n_z n_x}$ with $d_0 = -\frac{3}{2h_x}$, $d_1 = \frac{2}{h_x}$ and $d_2 = -\frac{1}{2h_x}$. The DtN maps in the left-hand side of (2.1d-e) act on the function values on the boundary only, i.e., the function approximated by $\hat{u}_\pm$. We compute the first p Fourier coefficients of the approximated function, apply the definition of $\mathcal{T}_{\pm,\gamma}$ on the Fourier coefficients, and convert the Fourier expansion back to the uniform grid. More precisely, the approximation of the left-hand side of (2.1)d-e is given by

$$\left[\mathcal{T}_{\pm,\gamma}(u(x_\pm, z))\big|_{z=z_i} \right]_{i=1}^{n_z} \approx R L_\pm(\gamma) R^{-1} \hat{u}_\pm \in \mathbb{C}^{n_z} \tag{2.10}$$

where $L_\pm(\gamma) = \text{diag}([s_{j,\pm}(\gamma)]_{j=-p}^p)$, and $R = [\exp(2i\pi j z_k)]_{k=1,j=-p}^{n_z,p}$ with $n_z = 2p + 1$. In the algorithm we exploit that the action of R and R^{-1} can be computed with FFT. We match (2.10) and (2.9) and get a discretization of the boundary condition (2.1)d-e. That is, we reach the NEP (1.4), with M given by (1.5) if we define $C_2 := \begin{bmatrix} C_{2,-} & C_{2,+} \end{bmatrix}$, $\Lambda_\pm(\gamma) := L_\pm + d_0 I$ and $w^T := \begin{bmatrix} \hat{u}^T & \hat{u}_-^T & \hat{u}_+^T \end{bmatrix}$ and combine (2.8) with (2.10) and (2.9).

3. Derivation and adaption of TIAR.

3.1. Basis matrix structure of the infinite Arnoldi method (IAR). There exist several variations of IAR, [16, 18]. We base our derivation of a variant of IAR in [18] called the Taylor variant, as it is based on the Taylor coefficients (derivatives) of

[1]For reproducability, we have provided MATLAB functions to generate the problem: `http://people.kth.se/~gmele/waveguide/`

M. Our algorithm can also be derived directly by using a function representation and infinite dimensional operators similar to [18]. Such a function setting was also natural in the derivation of restarting procedures for TIAR [25]. For reasons of conciseness, we derive our new algorithm by an equivalence with the Taylor variant. We now briefly summarize the algorithm and characterize a structure in the basis matrix. Similar to the standard Arnoldi method, IAR is an algorithm with an algorithmic state consisting of a basis matrix Q_k and a Hessenberg matrix H_k. The basis matrix and the Hessenberg matrix are expanded in every loop. Unlike the standard Arnoldi method, in IAR, the basis matrix is expanded by a block row as well as a column, leading to a basis matrix with block triangular structure, where the leading (top left) submatrix of the basis matrix is the basis matrix of the previous loop. More precisely, there exist vectors $q_{i,j} \in \mathbb{C}^n$, $i, j = 1, \ldots, k$ such that

$$Q_k = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,k} \\ 0 & q_{2,2} & & \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & q_{k,k} \end{bmatrix}. \tag{3.1}$$

In every loop in IAR we must compute a new vector to be used in the expansion of Q_k and H_k. In practice, in iteration k, this reduces to computing $y_1 \in \mathbb{C}^n$ given $y_2, \ldots, y_{k+1}$ such that

$$y_1 = -M(0)^{-1} \left(\sum_{i=1}^{k} M^{(i)}(0) y_{i+1} \right). \tag{3.2}$$

Clearly, since $M(0)$ does not change throughout the iteration, and we can compute an LU-factorization before starting the algorithm, such that the linear system can be solved efficiently in every iteration. IAR (Taylor version) is for completeness given by Algorithm 1.

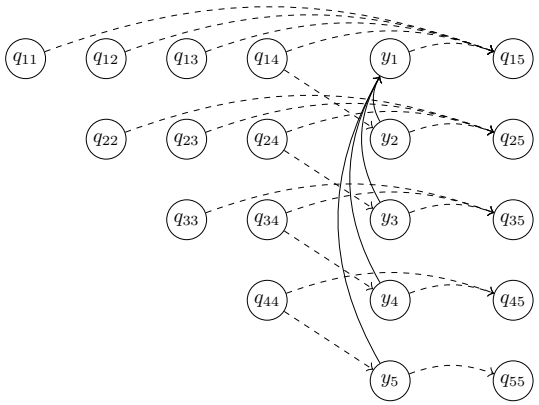


Figure 3.1: The computation tree representing Steps 3-9 for Algorithm 1 when $k = 4$, i.e., after three iterations. Every node is vector of size n. The dashed lines correspond to computing linear combinations. Clearly, the only (potentially) new direction of the span of all vectors can be represented by y_1.

Steps 3-9 of Algorithm 1 are visualized in Figure 3.1 when $k = 4$, i.e., after three iterations. We have marked those operations that are linear combinations as dashed

Algorithm 1: Infinite Arnoldi method - IAR (Taylor version) [18]

input : $x_1 \in \mathbb{C}^n$

1 Let $Q_1 = x_1/\|x_1\|_2$, H_0 =empty matrix
for $k = 1, 2, \ldots, m$ **do**
2 Compute $y_2, \ldots, y_{k+1}$ from the last column of Q_k by setting $y_j = \frac{1}{j-1} q_{j-1,k}$ for $j = 2, \ldots, k+1$.
3 Compute y_1 from $y_2, \ldots, y_{k+1}$ using (3.2)
4 Let $y := \mathrm{vec}(y_1, \ldots, y_{k+1})$ and $\underline{Q}_k := \begin{bmatrix} Q_k \\ 0 \end{bmatrix} \in \mathbb{C}^{(n(k+1))\times k}$.
5 Compute $h = \underline{Q}_k^H y$
6 Compute $y_\perp = y - \underline{Q}_k h$
7 Possibly repeat Steps 5-6 and get new h and $y_\perp$
8 Compute $\beta = \|y_\perp\|_2$
9 Compute $q_{k+1} = y_\perp/\beta$
10 Let $H_k = \begin{bmatrix} H_{k-1} & h \\ 0 & \beta \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}$
11 Expand Q_k into $Q_{k+1} = [\underline{Q}_k, q_{k+1}]$
end
12 Compute the eigenvalues $\{\mu_i\}_{i=1}^m$ of the leading $m \times m$ submatrix of the Hessenberg matrix H_k and return approximations $\{1/\mu_i\}_{i=1}^m$

lines. The fact that the many operations are linear combinations leads to a structure in Q_k which can be exploited such that we can reduce the usage of computer resources (memory and computation time) and maintain an equivalence with Algorithm 1.

More precisely, the block elements of the basis matrix Q_k have the following structure.

LEMMA 3.1 (Structure of basis matrix). *Let Q_i, $i = 1, \ldots, k$ be the sequence of basis matrices generated during the execution of $k-1$ iterations of Algorithm 1. Then, all block elements of the basis matrix Q_k (when partitioned as* (3.1)*) are linear combinations of $q_{1,1}, \ldots, q_{1,k}$.*

Proof. The proof is based on induction over the iteration count k. The result is trivial for $k = 1$. Suppose the results holds for some k. Due to the fact that Q_k is the leading submatrix of Q_{k+1}, as in (3.1), we only need to show that the blocks of the new column are a linear combinations $q_{i,j}$, $i, j = 1, \ldots, k$. This follows directly from the fact that $q_{2,k+1}, \ldots, q_{k+1,k+1}$ is (in step 3-9 in Algorithm 1) constructed as linear combination of $q_{1,k}, \ldots, q_{k,k}$. See Figure 3.1 □

We note that the structure presented in Lemma 3.1 is very natural in view of similar structures in other settings [21, Section 3.1], [39, Page 1057] and [36, Theorem 4.4]. To our knowledge, this has previously not been observed for IAR in the general case.

3.2. Derivation of TIAR. We now know from Lemma 3.1 that the basis matrix in IAR has a redundant structure. In this section we show that this structure can be exploited such that Algorithm 1 can be equivalently reformulated as an iteration involving a tensor factorization of the basis matrix without redundancy. We present a different formulation involving a factorization with a tensor which allows us to improve IAR both in terms of memory and computation time. This equivalent, but improved, version of Algorithm 1 appears to be competitive in general, and can be considerably

specialized to the waveguide eigenvalue problem as we show in section 4.

More precisely, Lemma 3.1 implies that there exists $a_{i,j,\ell}$ for $i,j,\ell = 1,\dots,k$ such that

$$q_{i,j} = \sum_{\ell=1}^{k} a_{i,j,\ell} z_\ell, \text{ for } i,j = 1,\dots,k \tag{3.3}$$

where $z_1,\dots,z_k$ is a basis of the span of the k first columns of the first block row, i.e., $\text{span}(q_{1,1},\dots,q_{1,k}) = \text{span}(z_1,\dots,z_k)$. Due to (3.3), the quantities $[z_1,\dots,z_k]$, $a_{i,j,\ell}$ for $i,j,\ell = 1,\dots,k$ can be interpreted as a factorization of Q_k. Rather than representing the basis directly by storing $q_{1,1},\dots,q_{1,k}$, we will use an orthogonal basis, $z_1,\dots,z_k$, i.e., the columns of the matrix $Z_k := [z_1,\dots,z_k] \in \mathbb{C}^{n\times k}$ are orthonormal. We use an orthogonal basis since we observed an improved numerical stability with an orthogonal basis. A similar difference in numerical stability has been completely characterized for similar methods for quadratic eigenvalues problems in [23]. Note that the first block row of Q_k can only be linearly independent if $k \le n$. This is the case for large-scale nonlinear eigenvalue problems, as the one we consider in this paper.

Suppose for the moment that we have carried out $k-1$ iterations of Algorithm 1. From Lemma 3.1 we know that the basis matrix can be factorized according to (3.3). The following results show that one loop, i.e., steps 2-11, can be carried out without explicitly storing Q_k, but instead only storing the factorization (3.3) represented by the tensor $a_{i,j,\ell}$ for $i,j,\ell = 1,\dots,k$ and the matrix $Z_k \in \mathbb{C}^{n\times k}$. Instead of carrying out operations on Q_k that lead to Q_{k+1}, we construct equivalent operations on the factorization of Q_k, i.e., $a_{i,j,\ell}$ for $i,j,\ell = 1,\dots,k$ and the matrix $Z_k \in \mathbb{C}^{n\times k}$, that directly lead to the factorization of Q_{k+1}, i.e., $a_{i,j,\ell}$ for $i,j,\ell = 1,\dots,k+1$ and the matrix $Z_{k+1} \in \mathbb{C}^{n\times(k+1)}$, without explicitly forming Q_k or Q_{k+1}.

To this end, suppose we have $a_{i,j,\ell}$ for $i,j,\ell = 1,\dots,k$ and $z_1,\dots,z_k$ available after $k-1$ iterations such that (3.3) is satisfied, and consider the steps 2-11 one-by-one. In Step 2 we need to compute the vectors $y_2,\dots,y_{k+1}$. They can be computed from the factorization of Q_k, since

$$y_j = \frac{1}{j-1} q_{j-1,k} = \frac{1}{j-1} \sum_{\ell=1}^{k} a_{j-1,k,\ell} z_\ell, \tag{3.4}$$

for $j = 2,\dots,k+1$. The vector y_1 is (in Step 3) computed using (3.2) and $y_2,\dots,y_{k+1}$ and does not explicitly require the basis matrix. For reasons of efficiency (which we further discuss in Remark 3.3) we carry out (3.4) with an equivalent matrix-matrix multiplication,

$$\begin{bmatrix}\tilde{y}_2 & \cdots & \tilde{y}_{k+1}\end{bmatrix} = Z_k A_k, \tag{3.5}$$

where $A_k^T = [a_{i,k,\ell}]_{i,j=1}^{k}$ and subsequently setting

$$y_j = \frac{1}{j-1}\tilde{y}_j \text{ for } j = 2,\dots,k+1. \tag{3.6}$$

In order to efficiently carry out the Gram-Schmidt orthogonalization process in step 5-9, it turns out to be efficient to first form a new vector z_{k+1}, which can be used in the factorized representation of Q_{k+1}. We *define* a new vector z_{k+1} via a Gram-Schmidt orthogonalization of y_1 against $z_1,\dots,z_k$. That is, we compute $z_{k+1} \in \mathbb{C}^n$ and $t_1,\dots,t_{k+1} \in \mathbb{C}$ such that

$$y_1 = t_1 z_1 + \cdots + t_{k+1} z_{k+1} \tag{3.7}$$

and expand $Z_{k+1} := \begin{bmatrix} Z_k & z_{k+1} \end{bmatrix}$ such that $Z_{k+1}^H Z_{k+1} = I$.

The new vector y (formed in Step 4) can now be expressed using the factorization, since

$$y = \begin{bmatrix} y_1 \\ \frac{1}{1} q_{1,k} \\ \frac{1}{2} q_{2,k} \\ \vdots \\ \frac{1}{k} q_{k,k} \end{bmatrix} = e_1 \otimes y_1 + \sum_{\ell=1}^{k} \begin{bmatrix} 0 \\ \frac{1}{1} a_{1,k,\ell} \\ \frac{1}{2} a_{2,k,\ell} \\ \vdots \\ \frac{1}{k} a_{k,k,\ell} \end{bmatrix} \otimes z_\ell = \sum_{\ell=1}^{k+1} \begin{bmatrix} g_{1,\ell} \\ \vdots \\ g_{k+1,\ell} \end{bmatrix} \otimes z_\ell \tag{3.8}$$

where we have defined $g_{i,\ell}$ as

$$g_{1,\ell} := t_\ell \text{ for } \ell = 1, \ldots, k+1 \tag{3.9a}$$

$$g_{i,\ell} := \frac{1}{i-1} a_{i-1,k,\ell} \text{ for } i = 2, \ldots, k+1, \quad \ell = 1, \ldots, k, \tag{3.9b}$$

$$g_{i,k+1} := 0 \text{ for } i = 2, \ldots, k+1. \tag{3.9c}$$

Instead of explicitly working with y, we store the matrix $[g_{i,\ell}]_{i,\ell=1}^{k+1} \in \mathbb{C}^{(k+1)\times(k+1)}$, representing the blocks of y as linear combinations of Z_{k+1}.

In order to derive a procedure to compute $h \in \mathbb{C}^k$ (in Step 5) without explicitly using Q_k, it is convenient to express the relation (3.3) using Kronecker products. We have

$$Q_k = \sum_{\ell=1}^{k} \begin{bmatrix} a_{1,1,\ell} & \cdots & a_{1,k,\ell} \\ \vdots & & \vdots \\ a_{k,1,\ell} & \cdots & a_{k,k,\ell} \end{bmatrix} \otimes z_\ell. \tag{3.10}$$

From the definition of h and (3.10) combined with (3.8) and the orthogonality of Z_{k+1}, we can now see that h can be expressed without explicitly using Q_k as follows

$$\begin{aligned} h = \underline{Q}_k^H y &= \left(\sum_{\ell=1}^{k} \begin{bmatrix} a_{1,1,\ell}^* & \cdots & a_{k,1,\ell}^* & 0 \\ \vdots & & \vdots & \vdots \\ a_{1,k,\ell}^* & \cdots & a_{k,k,\ell}^* & 0 \end{bmatrix} \otimes z_\ell^H \right) \left(\sum_{\ell'=1}^{k+1} \begin{bmatrix} g_{1,\ell'} \\ \vdots \\ g_{k+1,\ell'} \end{bmatrix} \otimes z_{\ell'} \right) \\ &= \sum_{\ell=1}^{k} \begin{bmatrix} a_{1,1,\ell}^* & \cdots & a_{k,1,\ell}^* \\ \vdots & & \vdots \\ a_{1,k,\ell}^* & \cdots & a_{k,k,\ell}^* \end{bmatrix} \begin{bmatrix} g_{1,\ell} \\ \vdots \\ g_{k,\ell} \end{bmatrix}. \end{aligned} \tag{3.11}$$

In Step 8 we need to compute the orthogonal completement of y with respect to $\underline{Q}_k$. This can be represented (without explicit use of Q_k) as follows

$$\begin{aligned} y_\perp = y - \underline{Q}_k h &= \sum_{\ell=1}^{k+1} \begin{bmatrix} g_{1,\ell} \\ \vdots \\ g_{k+1,\ell} \end{bmatrix} \otimes z_\ell - \sum_{\ell=1}^{k} \left(\begin{bmatrix} a_{1,1,\ell} & \cdots & a_{1,k,\ell} \\ \vdots & & \vdots \\ a_{k,1,\ell} & \cdots & a_{k,k,\ell} \\ 0 & \cdots & 0 \end{bmatrix} h \right) \otimes z_\ell \\ &= \sum_{\ell=1}^{k+1} \begin{bmatrix} f_{1,\ell} \\ \vdots \\ f_{k+1,\ell} \end{bmatrix} \otimes z_\ell, \end{aligned} \tag{3.12}$$

where we have used the elements of the matrix $[f_{i,j}]_{i,\ell=1}^{k+1}$ with columns defined by

$$\begin{bmatrix} f_{1,\ell} \\ \vdots \\ f_{k+1,\ell} \end{bmatrix} := \begin{bmatrix} g_{1,\ell} \\ \vdots \\ g_{k+1,\ell} \end{bmatrix} - \begin{bmatrix} a_{1,1,\ell} & \cdots & a_{1,k,\ell} \\ \vdots & & \vdots \\ a_{k,1,\ell} & \cdots & a_{k,k,\ell} \\ 0 & \cdots & 0 \end{bmatrix} h \tag{3.13}$$

for $\ell = 1, \ldots, k$ and $f_{i,k+1} := g_{i,k+1}$ for $i = 1, \ldots, k+1$.

We need β in Step 9, which is defined as the Euclidean norm of $y_\perp$. Due to the orthogonality of Z_{k+1}, we can also express β without using vectors of length n. In fact, it turns out that β is the Frobenius norm of the matrix $[f_{i,j}]_{i,j=1}^{k+1}$, since

$$\begin{aligned} \beta^2 = \|y_\perp\|^2 &= \left(\sum_{\ell=1}^{k+1} \begin{bmatrix} f_{1,\ell} \\ \vdots \\ f_{k+1,\ell} \end{bmatrix} \otimes z_\ell \right)^H \left(\sum_{\ell'=1}^{k+1} \begin{bmatrix} f_{1,\ell'} \\ \vdots \\ f_{k+1,\ell'} \end{bmatrix} \otimes z_{\ell'}, \right) \\ &= \sum_{\ell=1}^{k+1} \begin{bmatrix} f_{1,\ell} \\ \vdots \\ f_{k+1,\ell} \end{bmatrix}^H \begin{bmatrix} f_{1,\ell} \\ \vdots \\ f_{k+1,\ell} \end{bmatrix} = \left\| [f_{i,j}]_{i,j=1}^{k+1} \right\|_{\text{frob}}^2 . \end{aligned}$$

Finally (in Step 11), we expand Q_k by one column corresponding q_{k+1}, which is the normalized orthogonal complement. By using the introduced matrix $[f_{i,j}]_{i,j=1}^{k+1}$ we have that

$$q_{k+1} = \frac{1}{\beta} y_\perp = \frac{1}{\beta} \sum_{\ell=1}^{k+1} \begin{bmatrix} f_{1,\ell} \\ \vdots \\ f_{k+1,\ell} \end{bmatrix} \otimes z_\ell. \tag{3.14}$$

Let us now define

$$a_{i,k+1,\ell} = \frac{1}{\beta} f_{i,\ell}, \text{ for } i, \ell = 1, \ldots, k+1, \tag{3.15a}$$

$$a_{k+1,j,\ell} = 0 \text{ for } \ell = 1, \ldots, k+1, j = 1, \ldots, k \tag{3.15b}$$

$$a_{i,j,k+1} = 0 \text{ for } i = 1, \ldots, k+1, j = 1, \ldots, k. \tag{3.15c}$$

Hence, $a_{i,j,\ell}$ for $i, j, \ell = 1, \ldots, k+1$ and Z_{k+1} can be seen as a factorization of Q_{k+1} in the sense of (3.3), since the column added in comparison to the factorization of Q_k is precisely (3.14).

We summarize the above reasoning with a precise result showing how the dependence on Q_k for every step in Algorithm 1 can be removed, including how a factorization of Q_{k+1} can be constructed.

THEOREM 3.2 (Equivalent steps of algorithm). *Let Q_k be the basis matrix generated by $k-1$ iterations of Algorithm 1 and suppose $a_{i,j,\ell}$, for $i, j, \ell = 1, \ldots, k$ and Z_k are given such that they represent a factorization of Q_k of the type* (3.3). *The quantities computed (by executing Steps 2-11) in iteration k satisfy the following relations.*

(i) The vectors $y_2, \ldots, y_{k+1}$ computed in **Step 2**,*satisfy* (3.5).

(ii) Suppose y_1 (computed in Step 3) satisfies $y_1 \notin \operatorname{span}(z_1, \ldots, z_k)$. Let $z_{k+1} \in \mathbb{C}^n$ and $t_1, \ldots, t_{k+1} \in \mathbb{C}$ be the result of the Gram-Schmidt process satisfying (3.7). *Moreover, let $[g_{i,\ell}]_{i,\ell=1}^{k+1}$ be defined by* (3.9). *Then, then h computed in* **Step 5**, *satisfies* (3.11).

(iii) Let $[f_{i,\ell}]_{i,\ell=1}^{k+1}$ *be defined by* (3.13)*. Then, the vector* $y_\perp$*, computed in* **Step 6***, satisfies* (3.12)*.*

(iv) The scalar β*, computed in* **Step 8***, satisfies* $\beta = \left\| [f_{i,\ell}]_{i,\ell=1}^{k+1} \right\|_{\text{fro}}$

Moreover, if we expand $a_{i,j,\ell}$ *as in* (3.15)*, then,* $a_{i,j,\ell}$*, for* $i, j, \ell = 1, \ldots, k+1$ *and* $z_1, \ldots, z_{k+1}$ *represent a factorization of* Q_{k+1} *in the sense that* (3.3) *is satisfied for* $k+1$*.*

The above theorem directly gives us a practical algorithm. We state it explicitly in Algorithm 2. The details of the (possibly) repeated Gram-Schmidt process in Step 6–7 is straightforward and left out for brevity.

Algorithm 2: Tensor infinite Arnoldi method - TIAR

Input : $x_1 \in \mathbb{C}^n$

1 Let $Q_1 = x_1/\|x_1\|_2$, H_0 =empty matrix
for $k = 1, 2, \ldots, m$ **do**
2 Compute $y_2, \ldots, y_{k+1}$ from $a_{i,k,\ell}$, $i, \ell = 1, \ldots, k$ and Z_k using (3.5)-(3.6)
3 Compute y_1 from $y_2, \ldots, y_{k+1}$ using (3.2)
4 Compute $t_1, \ldots, t_{k+1}$ and z_{k+1} by orthogonalizing y_1 against $z_1, \ldots, z_k$ using a (possibly repeated) Gram-Schmidt process such that (3.7) is satisfied.
5 Compute the matrix $G = [g_{i,\ell}]_{i,\ell=1}^{k+1}$ using (3.9)
6 Compute $h \in \mathbb{C}^k$ using (3.11)
7 Compute the matrix $F = [f_{i,\ell}]_{i,\ell=1}^{k+1}$ using (3.13)
8 Possibly repeat Steps 6-7 and obtain updated h and F
9 Compute $\beta = \|F\|_{\text{fro}}$
10 Expand $a_{i,j,\ell}$ using (3.15)
11 Let $H_k = \begin{bmatrix} H_{k-1} & h \\ 0 & \beta \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}$
end
12 Compute the eigenvalues $\{\mu_i\}_{i=1}^m$ of the leading $m \times m$ submatrix of the Hessenberg matrix H_k and return approximations $\{1/\mu_i\}_{i=1}^m$

Remark 3.3 (Computational performance of IAR and TIAR). *Under the condition that* $q_{1,1}, \ldots, q_{1,m}$ *are linearly independent, Algorithm 1 (IAR) and Algorithm 2 (TIAR) are equivalent in exact arithmetic. The required computational resources of the two algorithms are however very different and TIAR appears to be preferable over IAR, in general.*

The first advantage of TIAR concerns the memory requirements. More precisely, in TIAR, the basis matrix is stored using a tensor $[a_{i,j,\ell}]_{i,j,\ell=1}^m \in \mathbb{C}^{m\times m\times m}$ *and a matrix* $Z_m \in \mathbb{C}^{n\times m}$*. Therefore, TIAR requires the storage of* $\mathcal{O}(m^3) + \mathcal{O}(mn)$ *numbers. In contrast to this, in IAR we need to store* $\mathcal{O}(m^2 n)$ *numbers since the basis matrix* Q_m *is of size* $mn \times m$*. Therefore, assuming that* $m \ll n$*, TIAR requires much less memory than IAR.*

The essential computational cost of carrying out m *steps of IAR consists of:* m *linear solves, computing* $\sum_{i=1}^k M^{(i)}(0)x_i$*, for* $k = 1, \ldots, m$*, and orthogonalizing a vector of length* kn *against* k *vectors of size* kn *for* $k = 1, \ldots, m$*. The orthogonalization has complexity*

$$t_{\text{IAR,orth}}(m, n) = \mathcal{O}(m^3 n), \tag{3.16}$$

which is the dominating cost when the linear solves are relatively cheap as in the waveguide eigenvalue problem.

On the other hand, the computationally dominating part of carrying out m steps of TIAR is as follows. Identical to IAR, m steps require m linear solves, and the computation of $\sum_{i=1}^{k} M^{(i)}(0)x_i$, for $k = 1, \ldots, m$. The orthogonalization process in TIAR (Step 4-9) is computationally cheaper than IAR. More precisely,

$$t_{\mathrm{TIAR,orth}}(m, n) = \mathcal{O}(m^2 n).$$

Unlike IAR, TIAR requires a computational effort in order to access the vectors $y_2, \ldots, y_k$ in Step 2 since they are implicitly given via $a_{i,j,k}$ and Z_k. In Step 2 we compute $y_2, \ldots, y_k$ with (3.6) *and* (3.5) *which correspond to multiplying a matrix of size $n \times k$ with a matrix of size $k \times k$ (and subsequently scaling the vectors). Hence, the operations corresponding to Step 2 for m iterations of TIAR can be carried out in*

$$t_{\mathrm{TIAR,Step\ 2}}(m, n) = \mathcal{O}(m^3 n). \tag{3.17}$$

At first sight, nothing is gained since the complexity of the orthogonalization in IAR (3.16) *and Step 2 of TIAR, are both $\mathcal{O}(m^3 n)$. However, it turns out that TIAR is often considerably faster in practice. This can be explained as follows. In the orthogonalization process of IAR we must compute $h = \underline{Q}_k^T y$ where $\underline{Q}_k \in \mathbb{C}^{kn \times k}$, whereas in Step 2 in TIAR we must compute $Z_k A_k^T$ (in* (3.5)*) where $Z_k \in \mathbb{C}^{n \times k}$ and $A_k \in \mathbb{C}^{k \times k}$. Note that the operation $Z_k A_k^T$ involves $nk + k^2$ values, whereas $\underline{Q}_k^T y$ involves nk^2 values, i.e., Step 2 in TIAR involves less data. This implies that on modern computer architectures, where CPU caching makes operations on small data-sets more efficient, it is in practice considerably faster to compute $Z_k A_k^T$ than $\underline{Q}_k^T y$ although the operations have the same computational complexity. This difference is also verified in the simulations in Section 5.*

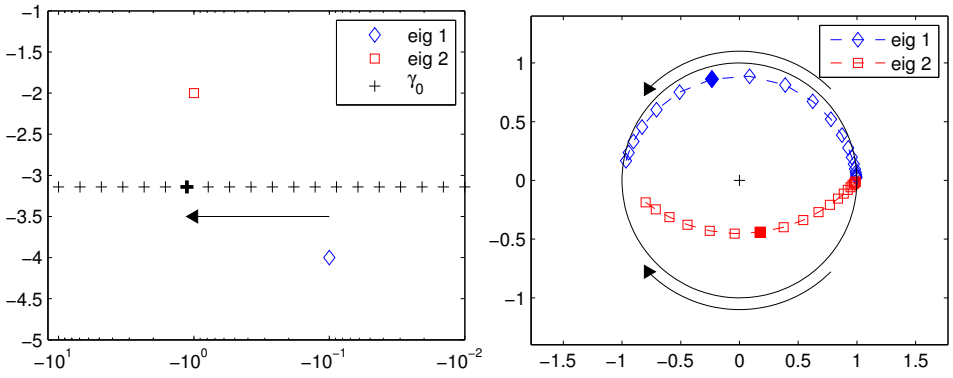
Although TIAR and IAR are mathemtically equivalent, they are not equivalent in finite arithmetic. As is pointed in [23] numerical stability can be an issue in this class of methods. The potential source of instability in SOAR discussed in [23] is however not present in our method, since we do not need to solve a linear system with a matrix containing coefficients. Moreover, in the context of our approach, we have not observed numerical instability and IAR and TIAR produce numerically similar eigenpair approximations.

4. Adaption to the waveguide problem.

4.1. Cayley transformation. One interpretation of IAR involves a derivation via the truncated Taylor series expansion. The truncated Taylor expansion is expected to converge slowly for points close to the branch-point singularities, and in general not converge at all for points further away from the origin than the closest singularity. Note that M, defined in (1.4), has branch point singularities at the roots of $\beta_{\pm,k}(\gamma)$ for $k = -p, \ldots, p$, where $\beta_{\pm,k}$ is defined in (2.3). In our situation, the eigenvalues of interest are close to the imaginary axis and, since the roots of $\beta_{\pm,k}$ are purely imaginary, the singularities are purely imaginary, which suggests poor performance of IAR (as well as TIAR) when applied to M.

In order to resolve this, we first carry out a Cayley transformation, which for a shift $\gamma_0 \in (-\infty, 0) \times (-2\pi, 0)i$ is given by

$$\lambda = \frac{\gamma - \gamma_0}{\gamma + \bar{\gamma}_0}, \tag{4.1}$$



(a) Before the Cayley transformation (b) After the Cayley transformation

Figure 4.1: Trajectories of the eigenvalues as a function of the shift γ_0.

and its inverse is $\gamma = (\gamma_0 + \lambda\bar{\gamma}_0)/(1-\lambda)$. The Cayley transformation moves the shift γ_0 to the origin, the singularities to the unit circle and the eigenvalues of interest to points inside the unit disk, i.e., inside the convergence disk. The choice of the shift γ_0 is problem dependent and may require different tries. In particular, the closer the shift is to the imaginary axis, the closer the transformed eigenvalues are to 1 and therefore to the border of the unitary disk. On the other hand, if the shift is far from the imaginary axis, the transformed eigenvalues will be localized at -1. An optimal choice of the shift moves the eigenvalues as far as possible from the border of the disk, therefore γ_0 should be selected in the middle of the region of interest with distance of the imaginary axis close to the expected location of the eigenvalues, see Figure 4.1.

Note that the transformed problem is still a nonlinear eigenvalue problem of the type (1.4), and we can easily remove the poles introduced by the denominator in (4.1). More precisely, we work with the transformed nonlinear eigenvalue problem

$$\begin{aligned}\tilde{M}(\lambda) &:= \begin{bmatrix}(1-\lambda)^2 I & \\ & (1-\lambda)I\end{bmatrix} M(\tfrac{\gamma_0+\lambda\bar{\gamma}_0}{1-\lambda}) \\ &= \begin{bmatrix} F_A(\lambda) & F_{C_1}(\lambda) \\ (1-\lambda)C_2^T & \tilde{P}(\lambda)\end{bmatrix},\end{aligned} \tag{4.2}$$

where

$$\begin{aligned} F_A(\lambda) &:= (1-\lambda)^2 A_0 + (\gamma_0+\lambda\bar{\gamma}_0)(1-\lambda)A_1 + (\gamma_0+\lambda\bar{\gamma}_0)^2 A_2, \\ F_{C_1}(\lambda) &:= (1-\lambda)^2 C_{1,0} + (\gamma_0+\lambda\bar{\gamma}_0)(1-\lambda)C_{1,1} + (\gamma_0+\lambda\bar{\gamma}_0)^2 C_{1,2}, \\ \tilde{P}(\lambda) &:= (1-\lambda)P(\frac{\gamma_0+\lambda\bar{\gamma}_0}{1-\lambda}). \end{aligned} \tag{4.3}$$

4.2. Efficient computation of y_1. In order apply IAR or TIAR to the waveguide problem, we need to provide a procedure to compute y_1 in Step 3 of Algorithm 1 and Algorithm 2 using (3.2). The structure of $\tilde{M}$ in (4.2) can be explicitly exploited and merged with Step 2 as follows. We analyze (3.2) for $k \geq 3$. It is straightforward to compute the corresponding formulas for $k < 3$. Due to the definition of $\tilde{M}$ in (4.2), formula (3.2) can be expressed as

$$-\tilde{M}(0)y_1 = z_1 + z_2, \tag{4.4}$$

with

$$z_1 := \begin{bmatrix} F_A'(0)y_{2,1} + F_{C_1}'(0)y_{2,2} + F_A''(0)y_{3,1} + F_{C_1}''(0)y_{3,2} \\ -C_2^T y_{2,1} \end{bmatrix},$$

$$z_2 := \begin{bmatrix} 0 \\ \sum_{i=1}^{k} \tilde{P}^{(i)}(0) y_{i+1,2}. \end{bmatrix} \tag{4.5}$$

where we have decomposed $y_i^T = [y_{i,1}^T, y_{i,2}^T]$, $i = 1, \ldots, k$ with $y_{i,1} \in \mathbb{C}^{n_x n_z}$ and $y_{i,2} \in \mathbb{C}^{2n_z}$. We solve the linear system (4.4) with a Schur complement. More precisely, we use that the (2,2)-block of (4.2) is explicitly available with FFT and we compute an LU-factorization of the Schur complement. In this way, computing solutions to (4.4) is not a dominating component (in terms of execution time) in our situation. The vector z_1 can be computed directly by using the definition of F_A and F_{C_1}. Using the definition of $\tilde{P}$ in (4.3), we can express the bottom block of z_2 as

$$z_{2,2} = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \sum_{i=1}^{k} D_i \left(\begin{bmatrix} R^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix} y_{i+1,2} \right) \in \mathbb{C}^{2n_z}, \tag{4.6}$$

where $D_i := \operatorname{diag}(\alpha_{-,-p,i}, \ldots, \alpha_{-,p,i}, \alpha_{+,-p,i}, \ldots, \alpha_{+,p,i})$ with

$$\alpha_{\pm,j,i} := \left(\frac{d^i}{d\lambda^i} \left((1-\lambda)(s_{\pm,j}(\frac{\gamma_0 + \lambda\bar{\gamma}_0}{1-\lambda}) + d_0) \right) \right)_{\lambda=0}. \tag{4.7}$$

In order to carry out m steps of the algorithm, we need to evaluate (4.7) $2n_z m$ times. We propose to do this with the efficient recursion formula given Appendix C. We note that similar formulas are used in [32] for slightly different functions.

Although the above formulas can be used directly to compute y_1, further performance improvement can be achieved by considerations of Step 2. Note that the complexity of Step 2 in TIAR is $\mathcal{O}(m^3 n)$, as given in equation (3.17). The computational complexity of this step can be decreased by using the fact that in order to compute y_1 in Step 3 and equation (4.4)-(4.5), we only need y_2, y_3 and $y_{4,2}, \ldots, y_{k+1,2} \in \mathbb{C}^{2n_z}$, i.e., not the full vectors. The structure can exploited in the operations in Step 2 as follows.

Let $B_{11} \in \mathbb{C}^{2\times 2}, B_{12} \in \mathbb{C}^{2\times(k-2)}, B_{21} \in \mathbb{C}^{(k-2)\times 2}$ and $B_{22} \in \mathbb{C}^{(k-2)\times(k-2)}$ be defined as blocks of A_k,

$$A_k = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}. \tag{4.8}$$

From (3.5) and (3.6) we have

$$\begin{bmatrix} \tilde{y}_2 & \tilde{y}_3 \end{bmatrix} = Z_k \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix}, \quad \begin{bmatrix} y_2 & y_3 \end{bmatrix} = \begin{bmatrix} \tilde{y}_2 & \tilde{y}_3/2 \end{bmatrix}, \tag{4.9}$$

and

$$\begin{bmatrix} \tilde{y}_{4,2} & \ldots & \tilde{y}_{k+1,2} \end{bmatrix} = Z_{k,2} B_{22}, \quad \begin{bmatrix} y_{4,2} & \ldots & y_{k+1,2} \end{bmatrix} = \begin{bmatrix} \tilde{y}_{4,2}/3 & \ldots & \tilde{y}_{k+1,2}/k \end{bmatrix}, \tag{4.10}$$

where $Z_{k,2} \in \mathbb{C}^{2n_z \times (k-2)}$ consists of the trailing block of Z_k.

By using formulas (4.9)-(4.10), we merge Step 2 and Step 3 in Algorithm 2 such that we can compute y_1 without computing the full vectors $y_2, \ldots, y_k$. For future reference we call this adaption WTIAR.

As explained in Remark 3.3, Step 2 of TIAR is the dominating component in terms of asymptotic complexity. With the adaption explained in (4.8)-(4.10), the complexity of Step 2 in WTIAR is

$$t_{\text{WTIAR,Step 2}}(m,n) = \mathcal{O}(nm^2) + \mathcal{O}(n_z m^3). \tag{4.11}$$

If the problem is discretized with the same number of discretization points in x-direction and z-direction, we have $t_{\text{WTIAR,Step 2}}(m,n) = \mathcal{O}(nm^2) + \mathcal{O}(\sqrt{n}m^3)$, which is considerable better than the complexity (3.17), i.e., the complexity of Step 2 in the plain TIAR. Notice that when n is sufficiently large the dominating term of the complexity of WTIAR is $\mathcal{O}(nm^2)$ which is also the complexity of the Arnoldi algorithm for the standard eigenvalue problem. The complexity is verified in practice in Section 5.

5. Numerical experiments.

5.1. Benchmark example. In order to illustrate properties of our approach, we consider a waveguide previously analyzed in [32, 6]. We set the wavenumber as in Figure 5.1, where $K_1 = \sqrt{2.3}\,\omega$, $K_2 = \sqrt{3}\,\omega$, $K_3 = \omega$ and $\omega = \pi$. Recall that the task is to compute the eigenvalues in the region $\Omega := (-\infty, 0) \times (-2\pi, 0)i \subset \mathbb{C}$, in particular those which are close to the imaginary axis.

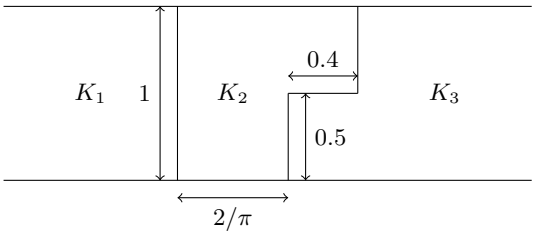


Figure 5.1: Geometry of the waveguide in Section 5.1

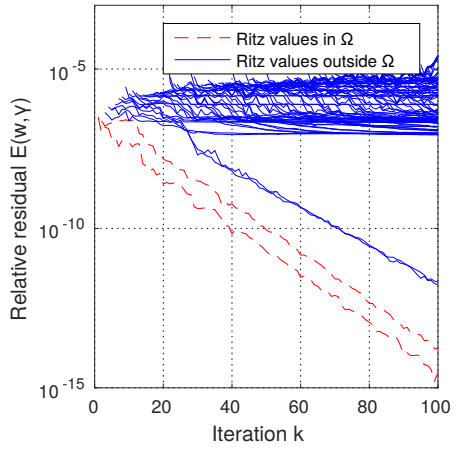
We select x_- and x_+ such that the interior domain is minimized, i.e., $x_- = 0$ and $x_+ = 2/\pi + 0.4$. The PDE is discretized with a FEM approach as explained in Section 2.2. Recall that the waveguide eigenvalue problem has branch point singularities and that the algorithms we are considering are based on derivations using Taylor series expansion. As explained in Section 4.1, the location of the shift γ in the Cayley transformation influences the convergence of the Taylor series, and cannot be chosen too close to the target, i.e., the imaginary axis. We select $\gamma_0 = -3i\pi$, i.e., in the middle of Ω in the imaginary direction. The error is measured using the relative residual norm

$$E(w,\gamma) := \frac{\|M(\gamma)w\|}{\sum_{i=0}^{2} |\gamma|^i \left(\|A_i\| + \|C_{1,i}\|\right) + \|C_2^T\| + 2d_0 + \sum_{j=-p}^{p} \left(|s_{j,+}(\gamma)| + |s_{j,-}(\gamma)|\right)}, \tag{5.1}$$

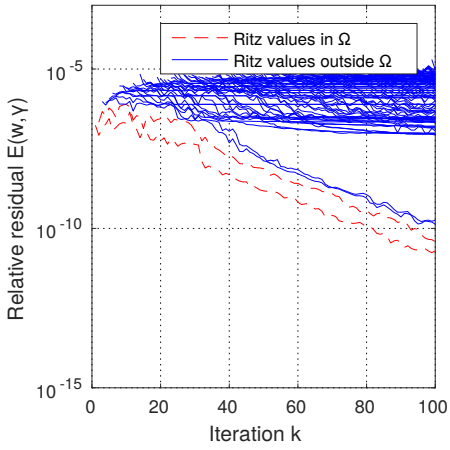
for $\|w\| = 1$.

We discretize the problem and we compute the eigenvalues of the nonlinear eigenvalue problem using WTIAR[2]. They are reported, for different discretizations, in Table 5.1. The required CPU time is reported in Table 5.2. The solution of the problem with the finest discretization, i.e., the last row of Table 5.1, was computed in more

[2]All simulations were carried out with Intel octa core i7-3770 CPU 3.40GHz and 16 GB RAM, except for the last two rows of Table 5.1 which were computed with Intel Xeon 2.0 GHz and 64 GB RAM.

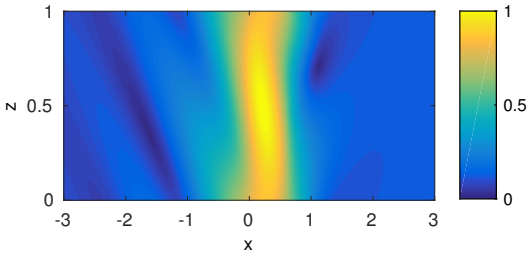


(a) Relative residual norm history $\gamma = -3-\pi i$

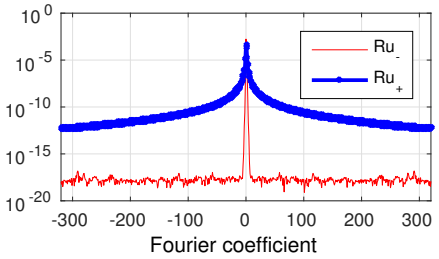


(b) Relative residual norm history $\gamma = -8-\pi i$

Figure 5.2: Execution of $m = 100$ iterations of WTIAR. The domain is discretized setting $n_x = 640$ and $n_z = 641$.



(a) Absolute value of the eigenfunction



(b) Fourier coefficients decay of $u(x_\pm, z)$

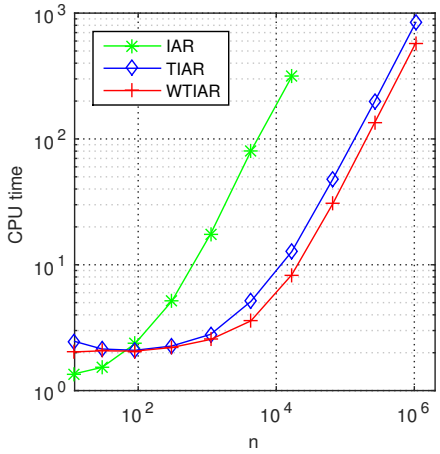
Figure 5.3: One solution of the waveguide eigenvalue problem with respect the waveguide in Figure 5.1. The domain is discretized setting $n_x = 640$ and $n_z = 641$.

than 10 hours. The bottleneck for the finest discretization is the memory requirements of the computation of the LU-factorization of the Schur complement corresponding to $\tilde{M}(0)$. An illustration of the execution of WTIAR, for this problem, is given in Figure 5.2, where the domain is discretized with $n_x = 640$ and $n_z = 641$ and $m = 100$ iterations are performed. We observe in Figure 5.2a and 5.2b that two Ritz values converge within the region of interest and two additional approximations converge to values with positive real part and of no interest in this application. The converge depends on the choice of the shift γ_0 as expected from the discussion in Section 4.1.

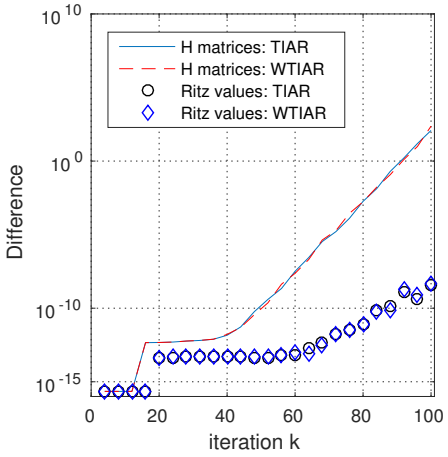
In Figure 5.3b we observe that the Fourier coefficients do not have exponential decay for $u(x_+, z)$. Indeed, the decay is quadratic, which is consistent with the fact that the solutions are C^1-functions, but in general not C^2, as explained in Remark 2.1. In particular, the second derivative of the eigenfunction is not continuous in $x = x_+$. Hence, the eigenfunctions appear to have just weak regularity, which means that the waveguide eigenvalue problem does not have a strong solution. This supports the choice of the discretization method, based on the FEM, that we use in this paper. In these simulations we selected $x_\pm$ such that the interior domain is minimized. We also carried out simulations for larger interior domains, without observing any qualitative difference

Problem size	n_x	n_z	First eigenvalue	Second eigenvalue
132	10	11	-0.010297987 - 4.966269257i	-0.008202089 - 1.390972357i
462	20	21	-0.009556975 - 4.965939619i	-0.009012367 - 1.337899343i
1,722	40	41	-0.009401369 - 4.965933116i	-0.009258151 - 1.322687924i
6,642	80	81	-0.009368285 - 4.966067569i	-0.009332752 - 1.318511833i
26,082	160	161	-0.009359775 - 4.966072322i	-0.009350769 - 1.317465909i
103,362	320	321	-0.009357649 - 4.966071811i	-0.009355348 - 1.317202268i
411,522	640	641	-0.009357159 - 4.966073495i	-0.009356561 - 1.317134070i
1,642,242	1,280	1,281	-0.009357028 - 4.966073418i	-0.009356859 - 1.317117443i
6,561,282	2,560	2,561	-0.009356994 - 4.966073409i	-0.009356933 - 1.317113346i
9,009,002	3,000	3,001	-0.009356991 - 4.966073406i	-0.009356938 - 1.317112905i

Table 5.1: Eigenvalue approximations stemming from WTIAR with $m = 100$.



(a) CPU time needed to perform $m = 100$ iterations of IAR, TIAR and WTIAR



(b) Difference between the matrices H_m and Ritz values in Ω computed with IAR and TIAR/WTIAR

Figure 5.4: Comparison of IAR, TIAR and the WTIAR in terms of CPU time and stability

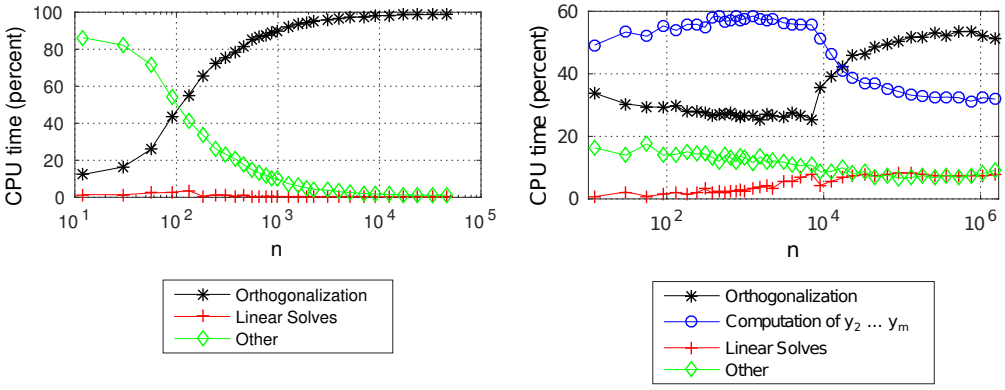
in the computed solutions. By Remark 2.1, this suggests that the C^1-assumption is not a restriction in this case.

The plot of the absolute value of one eigenfunction is given in Figure 5.3a. The convergence rate with respect to discretization, appears to be quadratic in the diameter of the elements. See Table 5.1.

As we mentioned in Remark 3.3, TIAR requires less memory and has the same complexity as IAR, although it is in practice considerable faster. According to Section 4.2, WTIAR requires the same memory resources as TIAR, but WTIAR has lower complexity. These properties are illustrated in Figure 5.4a and Table 5.2. As we showed in the theorem 3.2 TIAR and IAR are mathematically equivalent by construction. However, IAR and TIAR (as well as WTIAR) incorporate orthogonalization in different ways which may influence the impact of round-off errors. It turns out that the H_m matrices computed with IAR and TIAR are numerically different, but the Ritz values in Ω have a small difference. See Figure 5.4b. This suggests that there is an effect of the roundoff errors, but for the purpose of computing the Ritz values located in Ω, such error is not large for this problem.

			CPU time		storage of Q_m	
n	n_x	n_z	IAR	WTIAR	IAR	TIAR
462	20	21	8.35 secs	2.58 secs	35.24 MB	7.98 MB
1,722	40	41	28.90 secs	2.83 secs	131.38 MB	8.94 MB
6,642	80	81	1 min and 59 secs	4.81 secs	506.74 MB	12.70 MB
26,082	160	161	8 mins and 13.37 secs	13.9 secs	1.94 GB	27.52 MB
103,362	320	321	out of memory	45.50 secs	out of memory	86.48 MB
411,522	640	641	out of memory	3 mins and 30.29 secs	out of memory	321.60 MB
1,642,242	1280	1281	out of memory	15 mins and 20.61 secs	out of memory	1.23 GB

Table 5.2: CPU time and estimated memory required to perform $m = 100$ iterations of IAR and WTIAR. The memory requirements for the storage of the basis is the same TIAR and WTIAR.



(a) IAR (b) WTIAR

Figure 5.5: CPU time of the main parts of IAR and WTIAR with $m = 100$ iterations.

We mentioned in Section 4.2, that when n became sufficiently large, the dominating part of WTIAR is the orthogonalization process. This can be observed in Figure 5.5b. Recall that the orthogonalization in WTIAR has complexity $\mathcal{O}(nm^2)$, which is also the complexity of the standard Arnoldi algorithm. Hence, solving the waveguide eigenvalue problem with WTIAR using a fine discretization, has in this sense the same complexity as solving a standard eigenvalue problem of the same size using the Arnoldi algorithm. According to Remark 3.3 the dominating part of IAR is also the orthogonalization process, but this has higher complexity $\mathcal{O}(nm^3)$. See Figure 5.5a.

5.2. Waveguide with complex shape. In order to show the generality of our algorithm, we carried out simulations on a waveguide with a more complex geometry and solutions. It is described in Figure 5.6 where $K_1 = \sqrt{2.3}\omega$, $K_2 = 2\sqrt{3}\omega$, $K_3 = 4\sqrt{3}\omega$ and $K_4 = \omega$ and $\omega = \pi$.

We again select x_- and x_+ such that the interior domain is minimized, i.e., $x_- = 0$ and $x_+ = 2$. We discretize the problem and choose the same discretization parameters as in Section 5.1 and choose as shift $\gamma_0 = -2 - i\pi$. An illustration of the execution of WTIAR, for this problem, is given in Figure 5.7, where the domain is discretized with $n_x = 640$ and $n_z = 641$ and $m = 100$ iterations are performed. We observe that several Ritz values converge within the region of interest Ω. See Figure 5.7b and Figure 5.7a. One of the dominant eigenfunctions is illustrated in Figure 5.7c.

5.3. Simulations with a different method. In order to illustrate the competitiveness of our approach we present some simulations carried with a different method.

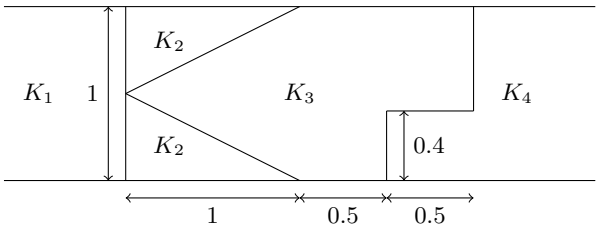
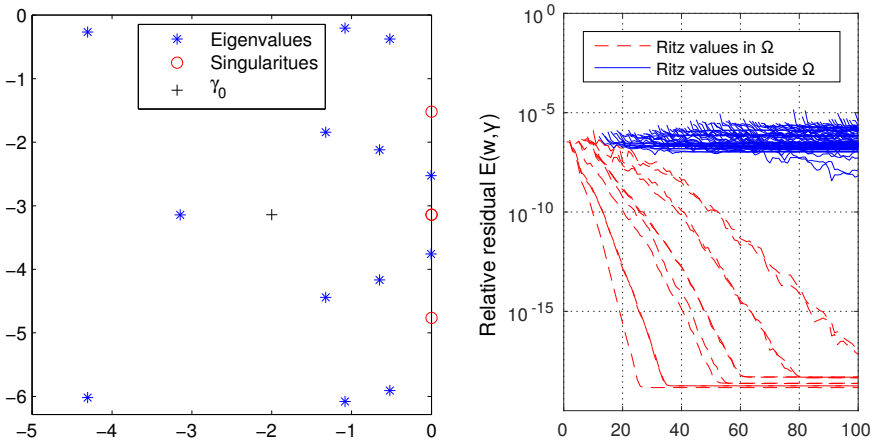
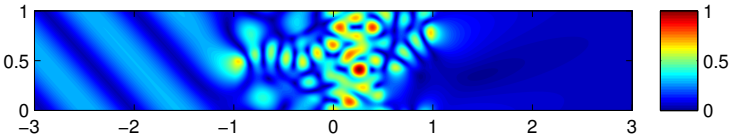


Figure 5.6: Geometry of the waveguide in Section 5.2



(a) Eigenvalues (converged Ritz values) and singularities

(b) Relative residual norm history

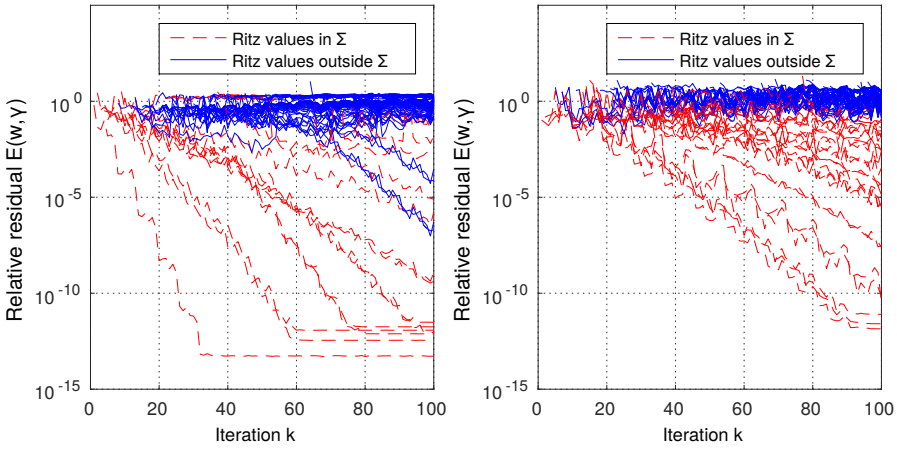


(c) Absolute value of the eigenfunction

Figure 5.7: Execution of $m = 100$ iterations of WTIAR. The parameters of the DtN maps are $x_- = 0$ and $x_+ = 2$. The domain is discretized setting $n_x = 640$ and $n_z = 641$.

We compare TIAR with NLEIGS as presented in [13] in terms of convergence rate (not CPU time). The compact representation results developed in [36, 33] can be adapted to improve the performance of NLEIGS, which however would not modify the convergence rate. We used the publicly available MATLAB implementation of NLEIGS provided by the same authors.

We briefly summarize the main features of NLEIGS. This algorithm is based on rational approximations of the nonlinear eigenvalue problem combined with a rational Krylov method for the linearized problem. Moreover, it is designed to compute eigenvalues in a region of interest Σ not containing singularities. The poles of the rational approximation are selected as a subset of the singularities of the problem, and the nodes are selected on $\partial\Sigma$ in a Leja–Bagby style. The algorithm has different



(a) Original problem (1.5) (b) Cayley transformed problem (4.2)

Figure 5.8: Residual history of $m = 100$ iterations of NLEIGS (static variant), where $x_- = 0$, $x_+ = 2$, $n_x = 100$ and $n_z = 101$.

variants which involve different shift selection strategies. In the dynamic and hybrid variants some of the shifts are coupled with the nodes, allowing a dynamic feature of the algorithm. In the static variant the shifts are selected independent of nodes. The static variant is a two-stage approach where the nonlinearity is approximated and subsequently the linearized problem is solved iteratively, i.e., a so called approximate-first approach.

We applied all three variants of NLEIGS to the waveguide eigenvalue problem, for the model in Section 5.2. Our problem has branch-point singularities, and therefore we selected Ξ as a discretization of the branch–cuts analogous to [13, Section 7.1]. The region Σ was selected as a rectangle containing all eigenvalues of interest, not including the imaginary axis. We also carried out simulations for the Cayley transformed problem (4.2), where Σ was selected as a disk of radius less than one. The dynamic and hybrid versions were in some situations able to compute some but never all eigenvalues of interest. In particular when the imaginary axis was almost included in Σ, the hybrid and dynamic variants of NLEIGS did not compute any eigenvalues for neither of the problems considered, whereas when Σ was selected sufficiently far from the imaginary axis, some eigenvalues were computed, which however were not all the eigenvalues of interest. With the static variant, NLEIGS was able to compute all solutions to a medium sized problem. The convergence diagram is presented in Figure 5.8. We observe that 100 iterations are insufficient to obtain good accuracy in this setting. In Figure 5.8, several of the eigenvalues of interest correspond to convergence curves with a larger convergence factor, and the eigenvalue closest to the imaginary axis has residual approximately 10^{-3}. In contrast to this, with our specialization of TIAR, all eigenvalues converged in 100 iterations. See Figure 5.7b.

We have here focused on a convergence rate comparison. A CPU time comparison would only be fair after incorporating focused research on a specialization of NLEIGS for this particular problem, and further tuning of method parameters. We limited our simulations to the ideas presented in [13] where the authors suggested to select the nodes in a Leja–Bagby style. This approach seems less effective than our approach to (1.5). Moreover, both algorithms (TIAR and NLEIGS) require the solutions of many linear systems involving the matrix $M(\bar{\lambda})$. As we mentioned in Section 5.1, for this

problem, the solution of such linear systems is a restriction of the problem size that can be solved with our approach. Since the same linear systems appear in NLEIGS, this restriction will also be present. Hence, even if a specialization of NLEIGS would be carried out, it will never be able be to solve larger problems than our approach.

6. Concluding remarks and outlook. In this paper we have presented a new general approach for NEPs and shown how to specialize the method to a specific problem stemming from analysis of wave propagation. Note that the non-polynomial nonlinearity arises from the absorbing boundary conditions. In our setting we were able to establish an explicit characterization of the DtN maps, which allowed us to incorporate the structure at an algorithmic level. This approach does not appear to be restricted to the waveguide problem. Many PDEs can be constructed with absorbing boundary conditions expressed in a closed form. By appropriate analysis, in particular differentiation with respect to the eigenvalue, the approach should carry over to other PDEs and other absorbing boundary conditions. Note that in our approach we selected x_- and x_+ such that the interior domain is minimized. If we select a larger domain, the decay of the Fourier coefficients in Figure 5.2c is faster and the DtN can be approximated with a γ-dependent low-rank matrix, for which other methods are available. In contrast to our approach, such an approach requires a larger interior domain to be discretized, a choice of x_- and x_+ and a choice of a truncation parameter.

There exist several variants of IAR, e.g., the Chebyshev version [18] and restarting variations [17]. There are also related rational Krylov methods [35]. The results of this paper may also be extendable to these situations, although this would require further analysis. In particular, all of these methods require (in some way) a quantity corresponding to formula y_1 in (3.2), for which the problem-dependent structure must be incorporated. The computation of this quantity must be accurate and efficient and require considerable problem-specific attention.

Acknowledgement. We thank Johan Karlsson for discussion and input regarding the Cayley transformation in Section 4.1 and the referees for careful reading of the manuscript and constructive comments.

Appendix A. Proof of Lemma 2.2. We consider the exterior problem on S_+ in (2.5). The proof corresponding to S_- is analogous. To simplify the notation we write $\beta_k = \beta_{+,k}(\gamma)$ and assume, without loss of generality, that $x_+ = 0$. By Remark 2.1 the solutions of (2.5) are in C^1 and every vertical trace can then be expanded in a Fourier series. Therefore we can express

$$w(x,z) = \sum_{k\in\mathbb{Z}} w_k(x) e^{2\pi i k z}, \qquad g(z) = \sum_{k\in\mathbb{Z}} g_k e^{2\pi i k z}.$$

By again using Remark 2.1, we have that the solutions to the exterior problem are in C^∞ if $x > 0$ and satisfy (2.5a). Therefore, the coefficients w_k satisfy

$$\sum_{k\in\mathbb{Z}} \left(w_k'' - (2\pi k)^2 w_k + 4\pi\gamma i w_k + (\gamma^2 + \kappa_+^2) w_k\right) e^{2\pi i k z} = \sum_{k\in\mathbb{Z}} \left(w_k'' + \beta_k w_k\right) e^{2\pi i k z} = 0,$$

where β_k is given in (2.3). Thus, in order for w to satisfy (2.5a), we have

$$w_k'' + \beta_k w_k = 0,$$

for all k. We now claim that there are constants C, C' independent of k such that

$$|\beta_k| \le C(1 + (2\pi k)^2), \qquad |\operatorname{Im}\sqrt{\beta_k}| \ge C'\sqrt{1 + (2\pi k)^2}. \tag{A.1}$$

In particular, $\beta_k \neq 0$ and

$$w_k(x) = c_k e^{i\text{sign}(\text{Im}(\beta_k))\sqrt{\beta_k}x} + d_k e^{-i\text{sign}(\text{Im}(\beta_k))\sqrt{\beta_k}x}.$$

To determine c_k and d_k we have two boundary conditions. First, since $w \in H^1(S_+)$, then $|w_k(x)|$ can not grow as $x \to \infty$. This means that $d_k = 0$. Second, at $x = 0$, we have $w(0, z) = g(z)$, so $c_k = g_k$. Hence, we have the explicit solution $w_k(x) = g_k e^{i\text{sign}(\text{Im}(\beta_k))\sqrt{\beta_k}x}$. Existence is thus proved, and the relationship (2.7) for the DtN maps follows directly for this solution by differentiating $w_k(x)$ and evaluating at $x = 0$. We also have $\mathcal{T}_{+,\gamma}[g] \in L^2(0, 1)$ since

$$||\mathcal{T}_{+,\gamma}[g]||^2_{L^2(0,1)} = \sum_{k\in\mathbb{Z}} |\beta_k|\,|g_k|^2 \leq C \sum_{k\in\mathbb{Z}} (1 + (2\pi k)^2)\,|g_k|^2 = C||g||_{H^1([0,1])}.$$

Finally, the estimate (2.6) is given by

$$\begin{aligned}
||w||^2_{H^1(S_+)} &= ||w||^2_{L^2(S_+)} + ||\nabla w||^2_{L^2(S_+)} = \sum_{k\in\mathbb{Z}} \Big[\big(1 + (2\pi k)^2\big)\,||w_k||^2_{L^2(0,\infty)} + ||w_k'||^2_{L^2(0,\infty)} \Big] \\
&= \sum_{k\in\mathbb{Z}} |g_k|^2 \left(1 + (2\pi k)^2 + |\beta_k|\right) \int_0^\infty e^{-2\text{sign}(\text{Im}(\beta_k))\,\text{Im}\,\sqrt{\beta_k}x} dx \\
&= \sum_{k\in\mathbb{Z}} \frac{|g_k|^2}{2|\,\text{Im}\,\sqrt{\beta_k}|} \left(1 + (2\pi k)^2 + |\beta_k|\right) \leq \frac{C+1}{2C'} \sum_{k\in\mathbb{Z}} |g_k|^2 \sqrt{1 + (2\pi k)^2} \\
&= \frac{C+1}{2C'} ||g||^2_{H^{1/2}([0,1])}.
\end{aligned}$$

Uniqueness follows from this estimate. It remains to show the claim (A.1). The estimate for $|\beta_k|$ is straightforward. For the second estimate we note that $|\,\text{Im}\,\beta_k| = 2|\,\text{Re}\,\gamma(2\pi k + \text{Im}\,\gamma)| \neq 0$ for all k from the assumptions $\text{Im}\,\gamma \notin 2\pi\mathbb{Z}$ and $\text{Re}\,\gamma \neq 0$. It follows that also $\text{Im}\,\sqrt{\beta}_k \neq 0$ for all k and since

$$\lim_{|k|\to\infty} a_k := \lim_{|k|\to\infty} \frac{\text{Im}\,\sqrt{\beta_k}}{\sqrt{1 + (2\pi k)^2}} = \text{Im}\,\sqrt{\lim_{|k|\to\infty} \frac{\beta_k}{1 + (2\pi k)^2}} = 1,$$

the sequence $\{1/a_k\}$ is bounded. Hence, there is a C' such that (A.1) holds, which concludes the proof.

Appendix B. Matrices of the FEM–discretization. The matrices $(A_i)_{i=0}^2$ and $(C_{1,i})_{i=0}^2$ are stem from to the Ritz–Galerkin discretization of the bilinear forms a, b and c. They can be decomposed and expressed as

$$\begin{aligned}
A_0 &:= D_{xx} + D_{zz} + K, & C_{1,0} &:= \tilde{D}_{xx} + \tilde{D}_{zz} + \tilde{K}, \\
A_1 &:= 2D_z, & C_{1,1} &:= 2\tilde{D}_z, \\
A_2 &:= B, & C_{1,2} &:= \tilde{B}.
\end{aligned}$$

Now we need to define the following tridiagonal Toeplitz matrices. Let E_m be the tridiagonal Toeplitz matrix with diagonals consisting of $e_{i+1,i} = -1$, $e_{i,i} = 2$ and $e_{i,i+1} = -1$. Let F_m be the tridiagonal Toeplitz matrix with diagonals consisting of $f_{i+1,i} = 1$, $f_{i,i} = 4$ and $f_{i,i+1} = 1$. Let G_m be the anti-symmetric tridiagonal Toeplitz matrix consisting of $g_{i+1,i} = 1$, $g_{i,i} = 0$ and $g_{i,i+1} = -1$.

Then, we have

$$D_{xx} = -\frac{h_z}{6h_x} E_{n_x} \otimes \left(B_{n_z} + e_{n_z} e_1^T + e_1 e_{n_z}^T \right), \quad \tilde{D}_{xx} = -\frac{h_z}{6h_x} - (e_1, e_{n_x}) \otimes \left(F_{n_z} + e_{n_z} e_1^T + e_1 e_{n_z}^T \right),$$

$$D_{zz} = -\frac{h_x}{6h_z} F_{n_x} \otimes \left(E_{n_z} - e_{n_z} e_1^T - e_1 e_{n_z}^T \right), \quad \tilde{D}_{zz} = -\frac{h_x}{6h_z} (e_1, e_{n_x}) \otimes \left(E_{n_z} - e_{n_z} e_1^T - e_1 e_{n_z}^T \right),$$

$$D_z = -\frac{h_x}{12} F_{n_x} \otimes \left(G_{n_z} + e_{n_z} e_1^T - e_1 e_{n_z}^T \right), \quad \tilde{D}_z = -\frac{h_x}{12} (e_1, e_{n_x}) \otimes \left(G_{n_z} + e_{n_z} e_1^T - e_1 e_{n_z}^T \right),$$

$$B = \frac{h_x h_z}{36} G_{n_x} \otimes \left(G_{n_z} + e_{n_z} e_1^T - e_1 e_{n_z}^T \right), \quad \tilde{B} = \frac{h_x h_z}{36} (e_1, -e_{n_x}) \otimes \left(G_{n_z} + e_{n_z} e_1^T - e_1 e_{n_z}^T \right).$$

The matrices K and $\tilde{K}$ arise from the Ritz–Galerkin discretization of the bilinear form

$$f(u, v) = \int_0^1 \int_{x_-}^{x_+} \kappa(x, z)^2 u(x, z) v(x, z) \, dx \, dz.$$

The elements of such matrices are obtained integrating the product of two basis functions against the square of the wavenumber. We can split the integral over the elements. Recall that $\kappa(x, z)$ is piecewise constant, then the final task is to compute, for each element, the integral of a piecewise polynomial function. Such integral is given in an explicit form by quadrature formulas.

Appendix C. Computation of the derivatives in DtN-map. In the computation of y_1 described in section 4.2, we need the coefficients $\alpha_{\pm,j,\ell}$. They can be computed with the following three-term recurrence.

LEMMA C.1 (Recursion for $\alpha_{\pm,j,\ell}$). *Suppose $\gamma_0 \notin i\mathbb{R}$. Then the coefficients in* (4.7) *are explicitly given by*

$$\begin{cases} \alpha_{\pm,j,0} = i w_j f_{\pm,j,0} + d_0 \\ \alpha_{\pm,j,1} = i w_j f_{\pm,j,1} - d_0 \\ \alpha_{\pm,j,\ell} = i w_j f_{\pm,j,\ell} \ell! \qquad \ell \geq 2, \end{cases} \tag{C.1}$$

where coefficients $f_{\pm,j,\ell}$ satisfy the following three-term recurrence

$$\begin{cases} f_{\pm,j,\ell} = -\dfrac{2a_{\pm,j}(\ell-3) f_{\pm,j,\ell-2} + b_{\pm,j}(2\ell-3) f_{\pm,j,\ell-1}}{2\ell c_{\pm,j}} \quad \ell \geq 2, \\ f_{\pm,j,0} = \sqrt{c_{\pm,j}}, \\ f_{\pm,j,1} = \dfrac{b_{\pm,j}}{2\sqrt{c_{\pm,j}}}, \end{cases} \tag{C.2}$$

with

$$\begin{cases} a_{\pm,j} = \bar{\gamma_0}^2 - 4\pi i j \bar{\gamma_0} - 4\pi^2 j^2 + \kappa_\pm^2, \\ b_{\pm,j} = 2\gamma_0 \bar{\gamma_0} + 4\pi i j (\bar{\gamma_0} - \gamma_0) + 8\pi^2 j^2 - 2\kappa_\pm^2, \\ c_{\pm,j} = 4\pi i j \gamma_0 - 4\pi^2 j^2 + \kappa_\pm^2 + \gamma_0^2, \\ w_j = \operatorname{sign}\left(\operatorname{Re}(\gamma_0)\left(\operatorname{Im}(\gamma_0) + 2\pi j\right)\right). \end{cases}$$

Proof. By definition (4.7)

$$\alpha_{\pm,j,\ell} = \left(\frac{d^\ell}{d\lambda^\ell} \left((1-\lambda) s_{\pm,j} \left(\frac{\gamma_0 + \lambda \bar{\gamma}_0}{1-\lambda} \right) \right) \right)_{\lambda=0} + \left(\frac{d^\ell}{d\lambda^\ell} \left((1-\lambda) d_0 \right) \right)_{\lambda=0}.$$

The computation of the second term is straightforward. The first term can be computed as follows. In order to compute the derivatives in zero, we now derive formulas for the Taylor expansion

$$(1-\lambda)s_{\pm,j}\left(\frac{\gamma_0+\lambda\bar{\gamma}_0}{1-\lambda}\right)=\sum_{\ell=0}^{+\infty}f_{\pm,j,\ell}\lambda^\ell.$$

Since all functions are analytic in the origin, there exists a neighborhood of the origin N, such that when $\lambda \in N$,

$$(1-\lambda)s_{\pm,j}\left(\frac{\gamma_0+\lambda\bar{\gamma}_0}{1-\lambda}\right)=w_j\sqrt{a_{\pm,j}\lambda^2+b_{\pm,j}\lambda+c_{\pm,j}}.$$

Hence, we have reduced the problem to computing the power series expansion of $\sqrt{a_{\pm,j}\lambda^2+b_{\pm,j}\lambda+c_{\pm,j}}$. To this end we use the well known formula involving the Gegenbauer polynomials and their generating function. See e.g. [27]. We have that

$$\begin{aligned}\sqrt{a_{\pm,j}\lambda^2+b_{\pm,j}\lambda+c_{\pm,j}} &= \sqrt{c_{\pm,j}}\left[\left(\sqrt{\frac{a_{\pm,j}}{c_{\pm,j}}}\lambda\right)^2+\frac{b_{\pm,j}}{\sqrt{a_{\pm,j}c_{\pm,j}}}\left(\sqrt{\frac{a_{\pm,j}}{c_{\pm,j}}}\lambda\right)+1\right]^{-\frac{1}{2}}\\ &= \sum_{\ell=0}^{+\infty}\sqrt{\frac{a_{\pm,j}^\ell}{c_{\pm,j}^{\ell-1}}}C_\ell^{(-1/2)}\left(-\frac{b_{\pm,j}}{2\sqrt{a_{\pm,j}c_{\pm,j}}}\right)\lambda^\ell,\end{aligned}$$

where C_ℓ is the ℓ-th Gegenbauer polynomial. Consequently, the coefficients in the power series expansion are

$$f_{\pm,j,\ell}=\sqrt{\frac{a_{\pm,j}^\ell}{c_{\pm,j}^{\ell-1}}}C_\ell^{(-1/2)}\left(-\frac{b_{\pm,j}}{2\sqrt{a_{\pm,j}c_{\pm,j}}}\right).$$

The recursion (C.2) follows from substitution of the recursion formula for Gegenbauer polynomials. □

REFERENCES

[1] Z. Bai, Y. Su, SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem, SIAM J. Matrix Anal. Appl. 26 (3) (2005) 640–659.

[2] G. Bao, Finite element approximation of time harmonic waves in periodic structures, SIAM J. Numer. Anal. 32 (4) (1995) 1155–1169.

[3] J.-P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. 114 (2) (1994) 185–200.

[4] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, F. Tisseur, NLEVP: A collection of nonlinear eigenvalue problems, ACM Trans. Math. Softw. 39 (2) (2013) 1–28.

[5] D. Bindel, S. Govindjee, Elastic PMLs for resonator anchor loss simulation, Int. J. Numer. Methods Eng. 64 (6) (2005) 789–818.

[6] J. Butler, W. Ferguson, G. A. Evans, P. J. Stabile, A. Rosen, A boundary element technique applied to the analysis of waveguides with periodic surface corrugations, IEEE J. of quantum electronics 28 (7) (1992) 1701–1709.

[7] C. Effenberger, D. Kressner, Chebyshev interpolation for nonlinear eigenvalue problems, BIT 52 (4) (2012) 933–951.

[8] C. Effenberger, D. Kressner, C. Engström, Linearization techniques for band structure calculations in absorbing photonic crystals, Int. J. Numer. Methods Eng. 89 (2) (2012) 180–191.

[9] C. Engström, Spectral approximation of quadratic operator polynomials arising in photonic band structure calculations, Numer. Math. 126 (3) (2014) 413–440.

[10] L. C. Evans, Partial differential equations. 2nd ed, American mathematical society, 2010.

[11] S. Fliss, A Dirichlet-to-Neumann approach for the exact computation of guided modes in photonic crystal waveguides, SIAM J. Sci. Comput. 35 (2) (2013) B438–B461.

[12] D. Givoli, I. Patlashenko, Dirichlet-to-Neumann boundary condition for time-dependent dispersive waves in three-dimensional guides, J. Comput. Phys. 199 (1) (2004) 339–354.

[13] S. Güttel, R. V. Beeumen, K. Meerbergen, W. Michiels, NLEIGS: a class of fully rational Krylov methods for nonlinear eigenvalue problems, SIAM J. Sci. Comput. 36 (6) (2014) A2842–A2864.
[14] T. Hagstrom, New results on absorbing layers and radiation boundary conditions, Ainsworth, Mark (ed.) et al., Topics in computational wave propagation. Direct and inverse problems. Berlin: Springer. Lect. Notes Comput. Sci. Eng.
[15] I. Harari, I. Patlashenko, D. Givoli, Dirichlet-to-Neumann maps for unbounded wave guides, J. Comput. Phys. 143 (1) (1998) 200–223.
[16] E. Jarlebring, K. Meerbergen, W. Michiels, A Krylov method for the delay eigenvalue problem, SIAM J. Sci. Comput. 32 (6) (2010) 3278–3300.
[17] E. Jarlebring, K. Meerbergen, W. Michiels, Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method, SIAM J. Matrix Anal. Appl. 35 (2) (2014) 411–436.
[18] E. Jarlebring, W. Michiels, K. Meerbergen, A linear eigenvalue algorithm for the nonlinear eigenvalue problem, Numer. Math. 122 (1) (2012) 169–195.
[19] L. Kaufman, Eigenvalue problems in fiber optic design, SIAM J. Matrix Anal. Appl. 28 (1) (2006) 105–117.
[20] J. B. Keller, D. Givoli, Exact non-reflecting boundary conditions, J. Comput. Phys. 82 (1) (1989) 172–192.
[21] D. Kressner, J. Roman, Memory-efficient Arnoldi algorithms for linearizations of matrix polynomials in Chebyshev basis, Numer. Linear Algebra Appl. 21 (4) (2014) 569–588.
[22] B.-S. Liao, Z. Bai, L.-Q. Lee, K. Ko, Nonlinear Rayleigh-Ritz iterative method for solving large scale nonlinear eigenvalue problems, Taiwanese Journal of Mathematics 14 (3) (2010) 869–883.
[23] D. Lu, Y. Su, Z. Bai, Stability analysis of the two-level orthogonal Arnoldi procedure, SIAM J. Matrix Anal. Appl. 37 (1) (2016) 195–214.
[24] V. Mehrmann, H. Voss, Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods, GAMM-Mitt. 27 (2004) 121–152.
[25] G. Mele, E. Jarlebring, Restarting for the tensor infinite Arnoldi method, Tech. rep., arXiv preprint arXiv:1606.08595, submitted (2016).
[26] S. T. Peng, T. Tamir, H. L. Bertoni, Theory of periodic dielectric waveguides, IEEE Trans. Microwave Theory and Techniques 1 (1975) 123–133.
[27] A. D. Polyanin, A. V. Manzhirov, Handbook of mathematics for engineers and scientists, CRC Press, 2006.
[28] D. Stowell, J. Tausch, Variational formulation for guided and leaky modes in multilayer dielectric waveguides, Commun. Comput. Phys. 7 (3) (2010) 564–579.
[29] Y. Su, Z. Bai, Solving rational eigenvalue problems via linearization, SIAM J. Matrix Anal. Appl. 32 (1) (2011) 201–216.
[30] Y. Su, J. Zhang, Z. Bai, A compact arnoldi algorithm for poly- nomial eigenvalue problems, presentation at the conference Recent Advances in Numerical Methods for Eigenvalue Problems (RANMEP2008), Taiwan, (2008).
[31] J. Tausch, Computing Floquet-Bloch modes in biperiodic slabs with boundary elements, J. Comput. Appl. Math. 254 (2013) 192–203.
[32] J. Tausch, J. Butler, Floquet multipliers of periodic waveguides via Dirichlet-to-Neumann maps, J. Comput. Phys. 159 (1) (2000) 90–102.
[33] R. Van Beeumen, Rational Krylov methods for nonlinear eigenvalue problems, Ph.D. thesis, KU Leuven (2015).
[34] R. Van Beeumen, E. Jarlebring, W. Michiels, A rank-exploiting infinite Arnoldi algorithm for nonlinear eigenvalue problems, Numer. Linear Algebra Appl. 23 (4) (2016) 607–628.
[35] R. Van Beeumen, K. Meerbergen, W. Michiels, A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems, SIAM J. Sci. Comput. 35 (1) (2013) A327–A350.
[36] R. Van Beeumen, K. Meerbergen, W. Michiels, Compact rational Krylov methods for nonlinear eigenvalue problems, SIAM J. Sci. Comput. 36 (2) (2015) 820–838.
[37] H. Voss, Nonlinear eigenvalue problems, in: L. Hogben (ed.), Handbook of Linear Algebra, Second Edition, No. 164 in Discrete Mathematics and Its Applications, Chapman and Hall/CRC, 2013.
[38] H. Voss, K. Yildiztekin, X. Huang, Nonlinear low rank modification of a symmetric eigenvalue problem, SIAM J. Matrix Anal. Appl. 32 (2) (2011) 515–535.
[39] Y. Zhang, Y. Su, A memory-efficient model order reduction for time-delay systems, BIT 53 (2013) 1047–1073.