Electronic Transactions on Numerical Analysis.
Volume 41, pp. 21-41, 2014.
Copyright © 2014, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

A SPATIALLY ADAPTIVE ITERATIVE METHOD FOR A CLASS OF NONLINEAR OPERATOR EIGENPROBLEMS[*]

ELIAS JARLEBRING[†] AND STEFAN GÜTTEL[‡]

Abstract. We present a new algorithm for the iterative solution of nonlinear operator eigenvalue problems arising from partial differential equations (PDEs). This algorithm combines automatic spatial resolution of linear operators with the infinite Arnoldi method for nonlinear matrix eigenproblems proposed by Jarlebring et al. [Numer. Math., 122 (2012), pp. 169–195]. The iterates in this infinite Arnoldi method are functions, and each iteration requires the solution of an inhomogeneous differential equation. This formulation is independent of the spatial representation of the functions, which allows us to employ a dynamic representation with an accuracy of about the level of machine precision at each iteration similar to what is done in the Chebfun system with its chebop functionality although our function representation is entirely based on coefficients instead of function values. Our approach also allows nonlinearity in the boundary conditions of the PDE. The algorithm is illustrated with several examples, e.g., the study of eigenvalues of a vibrating string with delayed boundary feedback control.

Key words. Arnoldi's method, nonlinear eigenvalue problems, partial differential equations, Krylov subspaces, delay-differential equations, Chebyshev polynomials

AMS subject classifications. 65F15, 65N35, 65N25

1. Introduction.

PDE eigenvalue problems arise naturally in many modeling situations. In some cases, e.g., when the PDE eigenvalue problem stems from a time-dependent PDE involving higher order derivatives in time or when it involves a delay, the corresponding PDE eigenvalue problem will be nonlinear in the eigenvalue parameter. In this paper we present a method for a class of PDE eigenvalue problems with that kind of nonlinearity. Examples of such problems are given, e.g., in [6, 8, 33] and in Section 4.

The nonlinear operator eigenvalue problem we are concerned with consists of finding a value $\lambda \in \mathbb{D}(\mu, r) := \{\lambda \in \mathbb{C} : |\lambda - \mu| < r\}$ close to $\mu \in \mathbb{C}$ and a nonzero function f such that

$$
\begin{aligned}
\mathcal{M}(\lambda) f &= 0, \\
c_1(\lambda, f) &= 0, \\
&\vdots \\
c_k(\lambda, f) &= 0.
\end{aligned}
\tag{1.1}
$$

In these equations, $\mathcal{M}(\lambda)$ denotes a family of operators defined on a common domain $D = D(\mathcal{M}(\lambda)) \subset L_2^w([a,b])$ and with a range space $L_2^w([a,b])$. The domain D here is assumed to be independent of the eigenvalue λ and will typically involve regularity conditions (e.g., differentiability). Note that for every fixed parameter λ, the operator $\mathcal{M}(\lambda)$ is linear but the dependence of $\mathcal{M}(\lambda)$ on λ is generally nonlinear. The set $L_2^w([a,b])$ denotes functions which are square integrable on the interval $[a,b]$ with a suitable weight function w. We shall specify a convenient weight function in Section 3 allowing us to efficiently compute scalar products in $L_2^w([a,b])$ numerically. The weight function is selected in order to achieve efficiency in the algorithm, and it does not necessarily correspond to the "natural inner product"

[*] Received November 27, 2012. Accepted October 29, 2013. Published online on March 17, 2014. Recommended by M. Hochstenbach. The work of S. G. was supported by Deutsche Forschungsgemeinschaft Fellowship GU 1244/1-1.

[†] Department of Mathematics, NA group, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (eliasj@kth.se).

[‡] The University of Manchester, School of Mathematics, Alan Turing Building, M13 9PL, Manchester, UK (stefan.guettel@manchester.ac.uk).

associated with physical properties of the involved operators. The functions $c_i : \mathbb{C} \times D \to \mathbb{C}$ specify k constraints that need to be satisfied for an eigenpair (λ, f).

We will assume that $\mathcal{M}(\lambda)$ can be represented as

$$\mathcal{M}(\lambda) = g_1(\lambda)\mathcal{L}_1 + g_2(\lambda)\mathcal{L}_2 + \cdots + g_m(\lambda)\mathcal{L}_m, \tag{1.2}$$

where $\mathcal{L}_i : D \to L_2^w([a,b])$ are closed linear operators and $g_i : \Omega \to \mathbb{C}$ are given analytic functions defined in an open neighborhood $\Omega \supset \mathbb{D}(\mu, r)$. We also assume that the constraints c_i can be represented in a similar fashion. More precisely, we assume that for all $i = 1, \ldots, k$ we have

$$c_i(\lambda, f) = h_{i,1}(\lambda)\mathcal{C}_{i,1}f + \cdots + h_{i,n}(\lambda)\mathcal{C}_{i,n}f,$$

where $h_{i,j} : \Omega \to \mathbb{C}$ are analytic functions and $\mathcal{C}_{i,j} : D \to \mathbb{C}$ are closed linear operators. We further assume that the constraints are such that the problem (1.1) is well posed in the sense that its solutions $\lambda \in \mathbb{D}(\mu, r)$ have finite multiplicities and are elements of a discrete set without accumulation points. The assumption that the spectrum is discrete restricts the problem class such that we do not face the complicated spectral phenomena that may occur for more general nonlinear operators; see, e.g., [1].

We have formulated the operator problem (1.1) in a quite general form, mostly for notational convenience. The problems we have in mind come from PDEs (with one spatial variable), e.g., PDEs with delays; see Section 4 for examples. For instance, the operators in (1.2) may correspond to differentiation

$$\mathcal{L}_1 = \frac{\partial}{\partial x}, \mathcal{L}_2 = \frac{\partial^2}{\partial x^2}, \ldots, \mathcal{L}_m = \frac{\partial^m}{\partial x^m}.$$

In this case, the functions c_i specify $k = m$ boundary conditions and we assume that they are such that (1.1) is a well-posed nonlinear operator eigenvalue problem.

The algorithm we propose is closely related to the *infinite Arnoldi method* presented in [19]. The infinite Arnoldi method can, in principle, solve nonlinear matrix eigenvalue problems (for eigenvalues in a disk) to arbitrary precision provided that certain derivatives associated with the problem are explicitly available. One can approach problems of the type (1.1) with the infinite Arnoldi method by first discretizing the PDE on the interval $[a, b]$, thereby obtaining a matrix eigenvalue problem whose solutions hopefully approximate those of (1.1). There are a number of approaches available for the nonlinear matrix eigenvalue problem [2, 4, 13, 25, 32]. Such a discretize-first approach requires an a priori choice of the discretization of the interval $[a, b]$. The algorithm presented here does not require such a choice because the spatial discretization will be adapted automatically throughout the iteration.

We derive the algorithm as follows. By approximating g_i and c_i by truncated Taylor expansions of order N, we first show that the resulting truncated operator eigenvalue problem can be written as an eigenvalue problem for an operator acting on arrays of functions in $L_2^w([a,b])^N$. This approach is similar to what for matrices is commonly called a companion linearization. See [24] for an analysis of companion linearizations. We select a particular companion-like operator formulation having a structure that is suitable for the Arnoldi method [28] applied to the operator formulation, and our derivation does not require a spatial discretization at this stage. We show that when the Arnoldi method for the companion-like operator formulation is initialized in a particular way, each iteration is equivalent to a result that would be obtained with an infinite truncation parameter N. We further exploit the structure of the Arnoldi method applied to the companion-like formulation so that the iterates of the algorithm are represented as arrays of $L_2^w([a,b])$ functions. The abstract algorithm presented in Section 2 can, in principle, find solutions to (1.1) with arbitrary accuracy with the main computational cost being the solution of an inhomogeneous differential equation derived from $\mathcal{M}(\mu)$ in every iteration.

As our algorithm derived in Section 2 is, in theory, based on iterating with functions in $L_2^w([a,b])$ and due to the fact that the algorithm does not involve a spatial discretization, we are free to choose the representation of the functions. In Section 3 we present an adaptive multi-level representation suitable to be combined with the algorithm in Section 2. Each iterate is represented via coefficients of a Chebyshev expansion of length automatically adapted to achieve machine precision. Details for some of the common $\mathcal{L}_i$-operations (like differentiation and pointwise multiplication) are also given in Section 3. In Section 4 we demonstrate the performance of our algorithm by three numerical examples.

Our approach of adaptive representation of functions together with an adaptive resolution of differential operators is clearly inspired by the Chebfun system [3] with its chebop functionality [12]. The idea to carry out iterations with functions has been presented in other settings. A variant of GMRES for functions is given in [26], where the functions are represented using Chebfun [3]. See also the discussion of infinite-dimensional numerical linear algebra in [17].

Apart from the notation introduced above, we use the following conventions. Calligraphic style will be used to denote operators, in particular, $\mathcal{I}$ will denote the identity operator and $\mathcal{O}$ will denote the zero operator. The set of one-dimensional (or k-dimensional) arrays of functions will be denoted by $L_2^w([a,b])^N$ (or $L_2^w([a,b])^{N\times k}$). The weighted 2-norm associated with a function $f \in L_2^w([a,b])^N$ will be denoted by $\|f\|_w$. The partial derivative with respect to λ will be denoted by $(\cdot)'$, the second partial derivative with respect to λ by $(\cdot)''$, etc.

2. The infinite Arnoldi method in an abstract PDE setting.

2.1. Truncated Taylor expansion. The derivation of our algorithm is based on a truncated Taylor-like expansion of the operator $\mathcal{M}$ around a given point $\mu \in \mathbb{C}$. Given an integer N, let the truncated operator $\mathcal{M}_N$ be defined by

$$\mathcal{M}_N(\lambda) := \mathcal{M}(\mu) + \frac{\lambda-\mu}{1!}\mathcal{M}^{(1)}(\mu) + \cdots + \frac{(\lambda-\mu)^N}{N!}\mathcal{M}^{(N)}(\mu),$$

with the operators $\mathcal{M}^{(j)}$ being analogues of the j-th derivative of $\mathcal{M}$ evaluated at μ,

$$\mathcal{M}^{(j)}(\mu) := g_1^{(j)}(\mu)\mathcal{L}_1 + g_2^{(j)}(\mu)\mathcal{L}_2 + \cdots + g_m^{(j)}(\mu)\mathcal{L}_m.$$

Accordingly, we define a Taylor-like expansion for the boundary conditions,

$$\begin{aligned} c_{i,N}(\lambda,f) := c_i(\mu,f) + \frac{\lambda-\mu}{1!}\left(\frac{\partial}{\partial\lambda}c_i(\lambda,f)\right)_{\lambda=\mu} + \\ \frac{(\lambda-\mu)^2}{2!}\left(\frac{\partial^2}{\partial\lambda^2}c_i(\lambda,f)\right)_{\lambda=\mu} + \cdots + \frac{(\lambda-\mu)^N}{N!}\left(\frac{\partial^N}{\partial\lambda^N}c_i(\lambda,f)\right)_{\lambda=\mu}. \end{aligned}$$

We now consider the truncated operator eigenproblem

$$\mathcal{M}_N(\lambda_N)f_N = 0, \tag{2.1a}$$

$$c_{i,N}(\lambda_N, f_N) = 0, \quad i = 1,\ldots,k \tag{2.1b}$$

with solution (λ_N, f_N). This eigenproblem approximates (1.1) in the sense that the residual of (λ_N, f_N) vanishes as $N \to \infty$. This is summarized in the following theorem.

THEOREM 2.1 (Convergence of operator Taylor-like expansion). *Let $\{(\lambda_N, f_N)\}_{N=1}^{\infty}$ denote a sequence of solutions to (2.1) with $f_N \in L_2^w([a,b])$ and $\lambda_N \in \mathbb{D}(\mu, r)$ for* *all $N = 1, 2\ldots$. Moreover, suppose that these solutions are convergent in the L_2^w norm, i.e., $(\lambda_N, f_N) \to (\lambda_*, f_*)$. Also suppose $\mathcal{L}_i f_N$ and $\mathcal{C}_{i,j} f_N$ are convergent in the L_2^w norm for any i, j as $N \to \infty$. Then there exist positive constants γ and $\beta < 1$ independent of N such that*

$$\begin{aligned}\|\mathcal{M}(\lambda_N)f_N\|_w &\le \gamma\beta^N\\ |c_1(\lambda_N, f_N)| &\le \gamma\beta^N\\ &\vdots\\ |c_k(\lambda_N, f_N)| &\le \gamma\beta^N.\end{aligned}$$

Proof. Since the functions g_i ($i = 1, \ldots, m$) are assumed to be analytic in a neighborhood of $\mathbb{D}(\mu, r)$, the complex Taylor theorem asserts that

$$g_i(\lambda) = \sum_{j=0}^{N} \frac{g_i^{(j)}(\mu)}{j!}(\lambda-\mu)^j + R_{i,N}(\lambda),$$

where the remainder term can be expressed via the Cauchy integral formula

$$R_{\ell,N}(\lambda) = \sum_{j=N+1}^{\infty} \frac{(\lambda-\mu)^j}{2\pi i}\int_\Gamma \frac{g_\ell(\zeta)}{(\zeta-\mu)^{j+1}}d\zeta$$

and Γ can be taken as a circular contour with center μ and radius $r > |\lambda - \mu|$. With the setting $M_{i,r} := \max_{\zeta\in\Gamma} |g_i(\zeta)|$, we obtain the standard Cauchy estimate

$$|R_{i,N}(\lambda)| \le \sum_{j=N+1}^{\infty} \frac{M_{i,r}|\lambda-\mu|^j}{r^j} \le \frac{M_{i,r}\hat\beta^{N+1}}{1-\hat\beta}$$

with $|\lambda - \mu|/r \le \hat\beta < 1$. Consequently,

$$\begin{aligned}\|\mathcal{M}(\lambda_N)f_N\|_w &= \|\mathcal{M}(\lambda_N)f_N - \mathcal{M}_N(\lambda_N)f_N\|_w\\ &= \|R_{1,N}(\lambda_N)\mathcal{L}_1 f_N + \cdots + R_{m,N}(\lambda_N)\mathcal{L}_m f_N\|_w\\ &\le \max_{i=1,\ldots,m} \frac{mM_{i,r}\hat\beta^{N+1}}{1-\hat\beta}\|\mathcal{L}_i f_N\|_w. \qquad (2.2)\end{aligned}$$

The conclusion about the bound on $\|\mathcal{M}(\lambda_N)f_N\|_w$ now follows from the fact that $\mathcal{L}_i f_N$ is assumed to be convergent. The conclusion about the bound on the boundary condition residuals follows from a completely analogous argument. The constants β and γ are formed by maximizing the computed bounds which are all of the form (2.2). □

REMARK 2.2. Theorem 2.1 illustrates that the residuals will decrease when N is sufficiently large and eventually approach zero as $N \to \infty$. The conclusion holds under the assumption that (λ_N, f_N) converges to a pair (λ_*, f_*). Despite this, note that the operators under consideration are not necessarily bounded, and therefore Theorem 2.1 does not necessarily imply that $\|\mathcal{M}(\lambda_*)f_*\|_w = 0$. For example, suppose that $\mathcal{M}(\lambda_*) = \frac{\partial}{\partial x}$ and consider a situation where (λ_N, f_N) is a solution to the truncated problem and $f_N(x) = f_*(x) + \frac{1}{N}\sin(Nx)$. Then $f_N \to f_*$ but $\mathcal{M}(\lambda_*)f_N$ will not converge to zero as $N \to \infty$. In such a situation, also a discretize-first approach could not be expected to give meaningful results. When f_* and all f_N are sufficiently smooth, this is unlikely to occur, and our numerical experiments in Section 4 suggest that such a situation would be rather artificial.

2.2. Operator companion linearization. From the above discussion it follows that one can approximate the original operator problem (1.1) by an operator problem where the coefficients in the operator and the boundary conditions are polynomials in λ. This is essentially an operator version of what is commonly called a *polynomial eigenvalue problem* [23, 29], and such problems are often analyzed and solved by the *companion linearization technique*. There are many types of companion linearizations [24], but for the purpose of this paper, a particular companion linearization is most suitable.

We first define an operator $\mathcal{A}_N$ acting on $L_2^w([a,b])^N$ such that

$$\mathcal{A}_N \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N \end{bmatrix} := \begin{bmatrix} \mathcal{M}(\mu) & & & \\ & \mathcal{I} & & \\ & & \ddots & \\ & & & \mathcal{I} \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N \end{bmatrix} = \begin{bmatrix} \mathcal{M}(\mu)\varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N \end{bmatrix} \tag{2.3}$$

and an operator $\mathcal{B}_N$ with action defined by

$$\mathcal{B}_N \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N \end{bmatrix} := \begin{bmatrix} -\mathcal{M}^{(1)}(\mu) & -\frac{1}{2}\mathcal{M}^{(2)}(\mu) & \cdots & \cdots & -\frac{1}{N}\mathcal{M}^{(N)}(\mu) \\ \mathcal{I} & \mathcal{O} & & & \\ & \frac{1}{2}\mathcal{I} & \ddots & & \\ & & \ddots & \ddots & \\ & & & \frac{1}{N-1}\mathcal{I} & \mathcal{O} \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N \end{bmatrix}. \tag{2.4}$$

Using these two operators, we can formulate the following generalized operator eigenproblem with boundary conditions

$$\mathcal{A}_N\varphi = (\lambda-\mu)\mathcal{B}_N\varphi \tag{2.5a}$$

$$\begin{aligned} &c_i(\mu,\varphi_1) + c_i'(\mu,\varphi_2) + \cdots \\ &\quad + c_i^{(N-1)}(\mu,\varphi_N) = -\frac{\lambda-\mu}{N} c_i^{(N)}(\mu,\varphi_N), \quad i = 1,\ldots,k. \end{aligned} \tag{2.5b}$$

This particular companion linearization is useful because, for any $M \geq N$, the leading $N \times N$ blocks in the operators $\mathcal{A}_M$ and $\mathcal{B}_M$ consist precisely of $\mathcal{A}_N$ and $\mathcal{B}_N$. This will be implicitly exploited in Section 2.3. The companion operator problem (2.5) is equivalent to the $\mathcal{M}_N$-problem (2.1) in the following sense.

THEOREM 2.3. *Consider $\varphi = (\varphi_1,\ldots,\varphi_N)^T \in L_2^w([a,b])^N$ with $\varphi_1 = f$. The companion linearization* (2.5) *and the truncated Taylor expansion* (2.1) *are equivalent in the sense that the following two statements are equivalent.*

a) The pair (λ,φ) is a solution to (2.5).

b) The pair (λ,f) is a solution to (2.1).

Proof. Consider a solution $\varphi = (\varphi_1,\ldots,\varphi_N)^T$ to (2.5). Then the last $N-1$ rows of (2.5a) imply that

$$\begin{aligned} \varphi_2 &= (\lambda-\mu)\varphi_1 \\ \varphi_3 &= \frac{1}{2}(\lambda-\mu)\varphi_2 = \frac{1}{2!}(\lambda-\mu)^2\varphi_1 \\ \varphi_4 &= \frac{1}{3}(\lambda-\mu)\varphi_3 = \frac{1}{3!}(\lambda-\mu)^3\varphi_1 \\ &\ \vdots \\ \varphi_N &= \frac{(\lambda-\mu)^{(N-1)}}{(N-1)!}\varphi_1. \end{aligned} \tag{2.6}$$

By inserting (2.6) into the first row in (2.5a), we have

$$\begin{aligned}0 &= \mathcal{M}(\mu)\varphi_1 + (\lambda-\mu)\mathcal{M}^{(1)}(\mu)\varphi_1 \\ &\quad + \frac{(\lambda-\mu)^2}{2!}\mathcal{M}^{(2)}(\mu)\varphi_1 + \cdots + \frac{(\lambda-\mu)^N}{N!}\mathcal{M}^{(N)}(\mu)\varphi_1.\end{aligned} \tag{2.7}$$

Similarly, (2.5b) implies with (2.6) and the linearity of $c_i(\lambda, f)$ with respect to f that

$$\begin{aligned}0 &= c_i(\mu,\varphi_1) + (\lambda-\mu)c_i'(\mu,\varphi_1) + \\ &\quad \frac{(\lambda-\mu)^2}{2!}c_i''(\mu,\varphi_1) + \cdots + \frac{(\lambda-\mu)^N}{N!}c_i^{(N)}(\mu,\varphi_1).\end{aligned} \tag{2.8}$$

The forward implication now follows from the fact that (2.7) is identical to (2.1a) and that (2.8) is identical to (2.1b).

In order to show the converse, suppose f is a solution to (2.1) and define $\varphi_1 = f$ and φ_i (for $i = 2, \ldots, N$) as in (2.6). The relation (2.7) holds because of (2.1), and a similar argument is used for the constraints (2.8). □

2.3. The infinite Arnoldi algorithm. Now note that (2.5) is a linear operator eigenproblem for the variable $\hat{\lambda} = (\lambda-\mu)^{-1}$. Linear eigenvalue problems can be solved in a number of ways, where the Arnoldi method [28] is one of the most popular procedures. We will now show how to formulate the Arnoldi method[1] for (2.5) and exploit the structure and thereby avoid the traditional approach to first discretize the problem. This is similar to the "Taylor version" of the infinite Arnoldi method for nonlinear matrix eigenvalue problems described in [19].

Conceptually, it is straightforward to use the Arnoldi method in an operator setting, and this has been done to study its convergence, e.g., in [9, 21]. In order to apply the Arnoldi algorithm to the formulation (2.5), we will need

- a procedure for solving

$$\mathcal{A}_N\varphi = \mathcal{B}_N\psi \tag{2.9a}$$

$$c_i(\mu,\varphi_1) + \cdots + c_i^{(N-1)}(\mu,\varphi_N) = -\frac{1}{N}c_i^{(N)}(\mu,\psi_N), \quad i = 1,\ldots,k \tag{2.9b}$$

for the unknown $\varphi \in L_2^w([a,b])^N$, where $\psi \in L_2^w([a,b])^N$ is given and
- a scalar product for $L_2^w([a,b])^N$.

It turns out that the structure of $\mathcal{A}_N$ and $\mathcal{B}_N$ is particularly well suited for the Arnoldi method. Suppose we start the Arnoldi method with a function $\psi \in L_2^w([a,b])^N$ of the form

$$\psi = \begin{bmatrix}\psi_1\\0\\\vdots\\0\end{bmatrix}, \tag{2.10}$$

where $\psi_1 \in L_2^w([a,b])$. In the first step of the Arnoldi method, we need to solve (2.9). By

[1] Note that our construction corresponds to a variant also known as shift-and-invert Arnoldi method since we actually approximate eigenvalues $\hat{\lambda} = \frac{1}{\lambda-\mu}$. For simplicity we will still refer to this variant as the Arnoldi method.

inspection of the structure of $\mathcal{A}_N$ and $\mathcal{B}_N$, the solution will be of the form

$$\varphi = \begin{bmatrix} \varphi_1 \\ \psi_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Hence, the action corresponding to the nonzero part of the solution of (2.9) is independent of N if we start with a vector consisting of just one leading nonzero block. More generally, the solution of (2.9) can be characterized as follows.

THEOREM 2.4. *Consider a given function $\psi \in L_2^w([a,b])^N$ with the structure*

$$\psi = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_p \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.11}$$

where $\psi_1, \ldots, \psi_p \in L_2^w([a,b])$. Consider the operators $\mathcal{A}_N$ and $\mathcal{B}_N$ defined by (2.3) *and* (2.4) *for any $N > p$. Suppose that $\varphi \in L_2^w([a,b])^N$ is a solution to the operator problem (in the space $L_2^w([a,b])^N$)*

$$\mathcal{A}_N \varphi = \mathcal{B}_N \psi \tag{2.12a}$$

$$c_i(\mu, \varphi_1) + \cdots + c_i^{(N-1)}(\mu, \varphi_{N-1}) = -\frac{1}{N} c_i^{(N)}(\mu, \psi_N), \quad i = 1, \ldots, k. \tag{2.12b}$$

Then this solution satisfies

$$\varphi = \begin{bmatrix} \varphi_1 \\ \frac{1}{1}\psi_1 \\ \vdots \\ \frac{1}{p}\psi_p \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.13}$$

where $\varphi_1 \in L_2^w([a,b])$ is the solution to the operator problem (in $L_2^w([a,b])$)

$$\mathcal{M}(\mu)\varphi_1 = -\mathcal{M}^{(1)}(\mu)\psi_1 - \frac{1}{2}\mathcal{M}^{(2)}(\mu)\psi_2 - \cdots - \frac{1}{p}\mathcal{M}^{(p)}(\mu)\psi_p \tag{2.14a}$$

$$c_i(\mu, \varphi_1) = -c_i'(\mu, \psi_1) - \frac{1}{2} c_i''(\mu, \psi_2) - \cdots - \frac{1}{p} c_i^{(p)}(\mu, \psi_p), \quad i = 1, \ldots, k. \tag{2.14b}$$

Proof. The last $N-1$ rows of (2.12a) imply that φ has the structure (2.13). Equation (2.14a) follows directly from the insertion of (2.13) and (2.11) into the first row of (2.12a). Note that the terms $\mathcal{M}^{(j)}(\mu)\psi_j$ vanish for $j > p$ since $\psi_j = 0$. Similarly, by inserting the structure of φ given in (2.13) and ψ given in (2.11) into Equation (2.12b), several terms vanish and (2.14b) is verified. □

From the previous theorem we make the following key observation.

> *The nonzero part of the solution to* (2.12) *for a function* ψ *with structure* (2.11) *is independent of* N *as long as* $N > p$.

By only considering functions of the structure (2.11) we can, in a sense, take $N \to \infty$ without changing the nonzero part of the solution. With $N \to \infty$, the truncation error in the Taylor expansion vanishes and (2.1) corresponds to the original problem (1.1) (under the conditions stated in Theorem 2.1 and Remark 2.2). In other words, our method has the remarkable property that at any iteration it gives the same results as if the Arnoldi method was run on the untruncated operator linearization. Hence, the truncation parameter can be formally considered as being $N = \infty$.

The key idea for an implementation is to start the Arnoldi algorithm with an array of functions of the structure (2.10). Due to the fact that the Arnoldi method essentially involves solutions of (2.12) at every iteration combined with a Gram–Schmidt orthogonalization, all arrays of functions will be of the structure (2.11). This naturally leads to a growth in the basis matrix in the Arnoldi algorithm not only by a column but also by a row at each iteration. The basis matrix after k iterations will be represented by

$$V = \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,k} \\ 0 & v_{2,2} & & \vdots \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & v_{k,k} \end{bmatrix} \in L_2^w([a,b])^{k\times k}, \tag{2.15}$$

where $v_{i,j} \in L_2^w([a,b])$.

In the Arnoldi algorithm we also need a scalar product. For the space $L_2^w([a,b])^N$ it appears to be natural to use the aggregated scalar product associated with a scalar product $\langle\cdot,\cdot\rangle_w$ for $L_2^w([a,b])$, i.e., given $f, g \in L_2^w([a,b])^N$, we define

$$\langle f, g\rangle_w := \langle f_1, g_1\rangle_w + \cdots + \langle f_N, g_N\rangle_w,$$

where $f = (f_1,\ldots,f_N)^T$, $g = (g_1,\ldots,g_N)^T$. The scalar product $\langle\cdot,\cdot\rangle_w$ can be tailored to the problem at hand, but we will propose a particularly convenient one in Section 3. A version of the Arnoldi algorithm that exploits the structure of the involved variables is given in Algorithm 1 below and referred to as the *infinite Arnoldi method (for nonlinear operator eigenproblems)*.

REMARK 2.5 (Existence). Algorithm 1 defines a sequence of function iterates uniquely only if there exists a unique solution to (2.14). Existence issues will not be studied in detail here and should be established in a problem specific manner. For the numerical examples we present in Section 4, existence and uniqueness of the solutions of (2.14) will be guaranteed by the well-posedness of the considered differential equations. The assumption that (2.14) has a solution in D, the domain of $\mathcal{M}$, is natural, though it is a restriction on the class of operator problems and allowed starting functions (which will be polynomials in our implementation, so this is not a practical restriction). Roughly speaking, this assumption means that only problems with sufficiently smooth solutions can be solved with our algorithm.

3. Multi-level spatial resolution.

The main computational cost in a practical implementation of our nonlinear eigensolver (Algorithm 1) lies in the solution of a differential equation (2.14) at every Arnoldi iteration. In this section we will propose a polynomial spectral method for solving differential equations with analytic (or sufficiently smooth) solutions defined on an interval $[a,b]$ suitable to be used in this setting. Because the Arnoldi method can

Algorithm 1 Infinite Arnoldi method for nonlinear operator eigenproblems (1.1).

```
Require: Starting function $v_{1,1} \in L_2^w([a,b])$
$v_{1,1} = v_{1,1}/\sqrt{\langle v_{1,1}, v_{1,1}\rangle_w}$
for $k = 1, 2, \ldots, k_{\max}$ do
    Compute $\varphi_2, \ldots, \varphi_{k+1}$ from (2.13) where $\psi_1 = v_{1,k}, \ldots, \psi_k = v_{k,k}$ and $p = k$.
    Solve the inhomogeneous differential equation (2.14) for $\varphi_1$ with the setting
    $\psi_1 = v_{1,k}, \ldots, \psi_k = v_{k,k}$ and $p = k$.
    for $i = 1, \ldots, k$ do
        $h_{i,k} = \langle \varphi_i, v_{1,i}\rangle_w + \cdots + \langle \varphi_i, v_{i,i}\rangle_w$
        for $j = 1, \ldots, i$ do
            $\varphi_j = \varphi_j - h_{i,k} v_{j,i}$
        end for
    end for
    $h_{k+1,k} = \sqrt{\langle \varphi_1, \varphi_1\rangle_w + \cdots + \langle \varphi_{k+1}, \varphi_{k+1}\rangle_w}$
    for $j = 1, \ldots, k+1$ do
        $v_{j,k+1} = \varphi_j / h_{k+1,k}$
    end for
end for
Compute the eigenvalues $\{\theta_i\}_{i=1}^{k_{\max}}$ of the Hessenberg matrix with elements $H_{i,j} = h_{i,j}$,
for $i, j = 1, \ldots, k_{\max}$.
Return the eigenvalue approximations $\{1/\theta_i + \mu\}_{i=1}^{k_{\max}}$ of (1.1).
```

be sensitive to inexact computations, we aim to solve these equations “exactly”, that is, with an error close to machine precision. Our approach is inspired by the automatic grid refinement idea implemented in the Chebfun system [3] with its chebop functionality [12], but it differs from Chebfun in the representation of the polynomials. The Chebfun system is based on interpolation polynomials represented on a Chebyshev grid with an adaptively chosen number of grid points, whereas we prefer to represent the polynomials by their coefficients in the Chebyshev basis. In other words, our approach is based on the *tau method* explained in Subsection 3.2 below instead of a collocation (or pseudospectral) method. The reason for our choice is that with a coefficient representation of polynomials, all operations required in our Arnoldi method can be implemented very efficiently without resampling function values between non-matching Chebyshev grids.

3.1. Coefficient spatial representation. Let $[a,b]$ be a given interval. In this section we will use the convention that with every occurrence of the variable x in $[a,b]$, we identify the variable $y = (2x - b - a)/(b - a)$ in $[-1,1]$. Any polynomial P_m of degree at most m can be represented as

$$P_m(x) = \sum_{j=0}^{m} c_j T_j(y), \quad x \in [a,b],$$

with the well-known Chebyshev polynomials $T_j(y) = \cos(j \arccos(y))$. Recall that these polynomials satisfy the recurrence

$$T_0(y) = 1, \quad T_1(y) = y, \quad T_{j+1}(y) = 2yT_j(y) - T_{j-1}(y),$$

and are orthogonal with respect to the weighted L_2^w scalar product

$$\langle f, g\rangle_w = \frac{2}{\pi}\int_{-1}^{1} \frac{f(y)\overline{g(y)}}{\sqrt{1-y^2}} dy,$$

more precisely

$$\langle T_j, T_k\rangle_w = \begin{cases} 0, & \text{if } j \neq k, \\ 2, & \text{if } j = k = 0, \\ 1, & \text{if } j = k \geq 1. \end{cases}$$

In contrast to the more popular spectral collocation approach [5, 11, 30], where a polynomial P_m is represented by its function values on a Chebyshev grid with nodes $y_j = \cos(\pi j/m)$ (for $j = 0, 1, \ldots, m$), we here prefer to represent P_m by its Chebyshev coefficients c_j. Given two polynomials $P_m(x) = \sum_{j=0}^m c_j T_j(y)$ and $Q_n(x) = \sum_{j=0}^n d_j T_j(y)$ of possibly different degrees, the coefficient representation allows us to compute linear combinations

$$\alpha P_m(x) + \beta Q_n(x) = \sum_{j=0}^{\max\{m,n\}} (\alpha c_j + \beta d_j) T_j(y),$$

without resampling function values of P_m or Q_n on a refined Chebyshev grid. (We assume that coefficients c_j or d_j with j exceeding the degree of the associated polynomial are equal to 0.) Moreover, it is easily verified that the Euclidean scalar product between coefficient vectors (with the 0-th coefficients divided by $\sqrt{2}$) corresponds to a weighted L_2^w scalar product between the corresponding polynomials:

$$\frac{c_0 \overline{d_0}}{2} + \sum_{j=1}^{\min\{m,n\}} c_j \overline{d_j} = \Big\langle \sum_{j=0}^m c_j T_j(y), \sum_{j=0}^n d_j T_j(y) \Big\rangle_w = \langle P_m, Q_n \rangle_w.$$

Note that our infinite Arnoldi method is rich in scalar product computations, and this relation allows for an efficient implementation.

3.2. The Chebyshev tau method with automated degree adaptation.

Given a polynomial P_m, in spectral methods one represents linear operations like differentiation $P_m \mapsto P_m'$, pointwise multiplication $P_m(x) \mapsto f(x)P_m(x)$, or the nonlocal reversal operation $P_m(x) \mapsto P_m(a+b-x)$ by matrix-vector products with spectral matrices. The *tau method* (invented by Lanczos [22], see also [5, Chapter 21], [18, Section 7.2]) is a spectral method for solving differential equations using the coefficient representation of polynomials where the coefficients are determined such that the residual of the approximate solution is orthogonal to as many basis polynomials as possible. The *Chebyshev tau method* is a tau method where the Chebyshev polynomials are used as a basis.

In the following we give an exemplary list of three coefficient maps representing the action of linear operators on a polynomial $P_m(x) = \sum_{j=0}^m c_j T_j(x)$. These maps will be needed in order to apply the algorithm to the examples in Section 4. For the identities involving Chebyshev polynomials used in the derivation, we refer to [14, Section 3].

- **Differentiation.** By the relation for the derivative of a Chebyshev polynomial $T_j(y)$,

$$\frac{d}{dy} T_j = \begin{cases} jT_0 + 2j(T_2 + T_4 + \cdots + T_{j-1}), & \text{if } j \text{ is odd}, \\ 2j(T_1 + T_3 + \cdots + T_{j-1}), & \text{if } j \text{ is even}, \end{cases}$$

we deduce that the matrix mapping the Chebyshev coefficients of P_m to the Chebyshev coefficients of P_m' is

$$D_m = \begin{bmatrix} 0 & 1 & 0 & 3 & 0 & \cdots \\ 0 & 0 & 4 & 0 & 8 & \cdots \\ 0 & 0 & 0 & 6 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 8 & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathbb{R}^{(m+1)\times(m+1)}.$$

Higher order derivatives are obtained by taking corresponding powers of the differentiation matrix D_m. Note that—in contrast to spectral collocation matrices acting on function values rather than coefficients—the matrix D_m is not dense.

- **Multiplication.** Let $Q_n(x) = \sum_{j=0}^{n} d_j T_j(y)$ be a polynomial. From the relation

$$T_j(y)T_k(y) = \frac{1}{2}\left(T_{j+k}(y) + T_{|j-k|}(y)\right),$$

it is easily verified that the matrix mapping the Chebyshev coefficients of P_m to the Chebyshev coefficients of P_mQ_n is

$$M_m(Q_n) = \frac{1}{2}\begin{bmatrix} d_0 & 0 & 0 & 0 & \cdots \\ d_1 & 2d_0 & d_1 & d_2 & \cdots \\ d_2 & d_1 & 2d_0 & d_1 & \cdots \\ d_3 & d_2 & d_1 & 2d_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} + \frac{1}{2}\begin{bmatrix} d_0 & d_1 & d_2 & d_3 & \iddots \\ d_1 & d_2 & d_3 & \iddots & \\ d_2 & d_3 & \iddots & & \\ d_3 & \iddots & & & \\ \iddots & & & & \end{bmatrix} \in \mathbb{C}^{(m+n+1)\times(m+n+1)},$$

which is the sum of a rank-1-modified Toeplitz matrix and a Hankel matrix.

- **Reversal.** Using the fact that $T_j(y) = (-1)^j T_j(-y)$, it is easily verified that the matrix

$$R_m = \operatorname{diag}(1, -1, 1, -1, \ldots) \in \mathbb{R}^{(m+1)\times(m+1)}$$

maps the coefficients of $P_m(x)$ to the coefficients of the "reversed" (right-to-left) polynomial $P_m(a+b-x)$.

- **Combinations of the above.** Note that the above operators can be combined in an additive and multiplicative fashion by adding and multiplying the corresponding matrices. For example, the variable coefficient second-order operator $\frac{d}{dy}(Q(y)\frac{d}{dy}\,\cdot)$ can be approximated as $D_{m+n}M_m(Q_n)D_{m+n}$ provided that $Q(y)$ can be (uniformly) approximated by a Chebyshev expansion Q_n of moderate degree n. For nonsmooth functions $Q(y)$, however, a global Chebyshev expansion may fail to converge (e.g., in the case of jumps causing the Gibbs phenomenon) or converge slowly (e.g., in the case of discontinuous derivatives); see [5, 30, 31]. Both of these cases would require a more sophisticated approach, such as, e.g., piecewise polynomial representations.

Let A be a linear operator acting on functions defined on the interval $[a, b]$, and denote by $A_m \in \mathbb{C}^{(m+1)\times(m+1)}$ the spectral matrix mapping the Chebyshev coefficients of polynomials P_m to the Chebyshev coefficients of $Q_m = AP_m$,

$$\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_m \end{bmatrix} = A_m \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{bmatrix}.$$

(Again we have assumed that coefficients with an index exceeding the degree of a polynomial are set to 0.) Typically the matrix A_m is not invertible. In order to specify P_m uniquely as the solution of the linear system $A_m P_m = Q_m$ for a given right-hand side Q_m, a number of constraints, say k, need to be imposed on P_m. In the tau method this is typically achieved by replacing the last k rows of A_m by row vectors corresponding to the boundary conditions of the differential equation (boundary bordering), e.g.,

$$\begin{aligned} \left(1, -1, 1, -1, \dots, (-1)^{m+1}\right) &\quad \text{Dirichlet b.c. on the left} \\ \left(1, 1, \dots, 1\right) &\quad \text{Dirichlet b.c. on the right} \\ \tfrac{2}{b-a}\left(0, 1, -2, 4, \dots, (-1)^m (m-1)^2\right) &\quad \text{Neumann b.c. on the left} \\ \tfrac{2}{b-a}\left(0, 1, 2, 4, \dots, (m-1)^2\right) &\quad \text{Neumann b.c. on the right,} \end{aligned}$$

and to alter the last k coefficients of Q_m, namely $(d_{m-k+1}, \dots, d_m)^T$, to the prescribed boundary values (zeros for homogeneous conditions). The results of this modification are denoted as $\underline{A_m}$ and $\underline{Q_m}$, respectively. This ensures that the polynomial $P_m = \underline{A_m}^{-1}\underline{Q_m}$ satisfies the boundary conditions exactly and the residual for the original differential operator is of the form

$$Q_m(x) - AP_m(x) = \sum_{j=m+1-k}^{\infty} e_j T_j(y)$$

provided that the exact solution $A^{-1}Q_m$ exists and has a Chebyshev expansion. Lanczos realized that with P_m, we have obtained the exact polynomial solution of $AP_m = Q_m + \epsilon_m$ to a (slightly) perturbed problem, $\epsilon_m = -\sum_{j=m+1-k}^{\infty} e_j T_j(y)$. Under the condition that P_m converges uniformly to a solution function f (the solution of the spectrally discretized differential equation) as $m \to \infty$ and the condition that this function f is analytic in a neighborhood of the interval $[a, b]$ (the *Bernstein ellipse*), it is known that the convergence is geometric (see, e.g., [31, Chapter 8]): for some $\rho > 1$ and $C > 0$, one has

$$|f(x) - P_m(x)| \le C\rho^{-m} \quad \text{for all } x \in [a, b].$$

If f has no singularities too close to $[a, b]$, then ρ is large enough to achieve fast uniform convergence of P_m towards f, indicated by a rapid decay of P_m's Chebyshev coefficients $c_0, c_1, \dots, c_m$. This fact is exploited in the Chebfun system with its chebop functionality for solving operator equations [12], and we will employ a similar rule of thumb: assume that the weighted L_2^w error of a Chebyshev approximant P_m is of about the same order as its trailing Chebyshev coefficient c_m (see also [5, p. 51]). This error estimate allows us to adaptively adjust the degree of P_m such that the solution P_m of $A_m P_m = Q_m$ is likely to be close to $A^{-1}f$ in a relative error sense:

1. Choose a number m, say $m = 16$.
2. Construct $\underline{A_m}$ (the spectral matrix with boundary conditions included), and solve the linear system $\underline{A_m} P_m = \underline{Q_m}$ for $P_m(x) = \sum_{j=0}^{m} c_j T_j(x)$.
3. If the last coefficient $c_m / \|P_m\|_w$ is not small enough relative to the norm $\|P_m\|_w$ induced by $\langle \cdot, \cdot \rangle_w$, increase m (e.g., multiply by a factor of 1.5 and round to integer), and go to Step 2.

Note that more sophisticated error estimates could be developed (for example, by taking into account more than just the last Chebyshev coefficient c_m). However, every such estimate will eventually be based on a heuristic. In the numerical experiments described in Section 4, we found the above procedure (Steps 1–3) to perform satisfactorily.

3.3. Implementation.

The implementation of our infinite Arnoldi method is straightforward in object-oriented Matlab. All spatial functions $v_{i,j}$ defined on the interval $[a, b]$ are approximated by polynomials $P_{i,j}$ of degree adaptively chosen such that the estimate $\|v_{i,j} - P_{i,j}\|_w \lessapprox \texttt{tol}\|P_{i,j}\|_w$ holds, where $\texttt{tol} = 2.2 \times 10^{-16}$. These polynomial representations are stored in a two-dimensional “cell array” (cf. (2.15))

$$V = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & \cdots \\ P_{2,1} & P_{2,2} & P_{2,3} & \cdots \\ P_{3,1} & P_{3,2} & P_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where each column corresponds to a Krylov basis vector and V will have an upper triangular structure. The action of the linear companion operator onto a column of V results in a new column of spatial functions, where the number of nonzero components in the input and output columns may be different. Note that a modified Gram–Schmidt orthogonalization of these columns is fast when working with the coefficient representation described above.

4. Examples.

4.1. A differential equation with time delay.

We consider a PDE with delay for a function $u : [0, \pi] \times [-\tau, +\infty) \to \mathbb{R}$,

$$u_t(x,t) = u_{xx}(x,t) - u(x, t-\tau), \tag{4.1a}$$

$$u(0,t) = 0, \tag{4.1b}$$

$$u(\pi,t) = 0, \tag{4.1c}$$

an example which has also been considered in [7, Formula (112)]. Employing the ansatz $u(x,t) = f(x)e^{\lambda t}$, the PDE (4.1) leads to a nonlinear operator eigenvalue problem of the form (1.1), where

$$\mathcal{M}(\lambda) = -\lambda \mathcal{I} + \frac{\partial^2}{\partial x^2} - e^{-\tau\lambda}\mathcal{I}, \tag{4.2}$$

with boundary conditions

$$c_1(\lambda, f) = f(0),$$
$$c_2(\lambda, f) = f(\pi).$$

In the implementation of our method we need to provide the derivatives of (4.2), which in this case are explicitly given by

$$\begin{aligned}
&\mathcal{M}^{(1)}(\mu) = -\mathcal{I} + \tau e^{-\tau\mu}\mathcal{I}\\
&\mathcal{M}^{(k)}(\mu) = (-\tau)^k e^{-\tau\mu}\mathcal{I}, \quad k \geq 2.
\end{aligned}$$

Consequently, in every iteration of our algorithm we need to solve (2.14), which reduces to solving

$$\begin{aligned}
&\left(-\mu\mathcal{I} + \frac{\partial^2}{\partial x^2} - e^{-\tau\mu}\mathcal{I}\right)\varphi_1\\
&\qquad = (1+\tau e^{-\tau\mu})\psi_1 - \frac{1}{2}(-\tau)^2 e^{-\tau\mu}\psi_2 - \cdots - \frac{1}{p}(-\tau)^p e^{-\tau\mu}\psi_p
\end{aligned}$$

for φ_1 with boundary conditions $\varphi_1(0) = \varphi_1(\pi) = 0$.

In this first example we have selected $\mathcal{M}(\lambda)$ such that the problem can be solved explicitly as follows. By defining $\gamma := \lambda + e^{-\tau\lambda}$, it is clear from (4.2) that all such γ correspond to the eigenvalues of the Laplacian with homogeneous boundary conditions, i.e., $\frac{\partial^2}{\partial x^2} f = \gamma f$ with $c_1(\lambda, f) = f(0) = 0$, $c_2(\lambda, f) = f(\pi) = 0$. This eigenvalue problem can be solved analytically and the explicit eigenfunction solution is $f(x) = \sin(jx)$ with eigenvalues $\gamma = -j^2$ for any positive integer j. Hence,

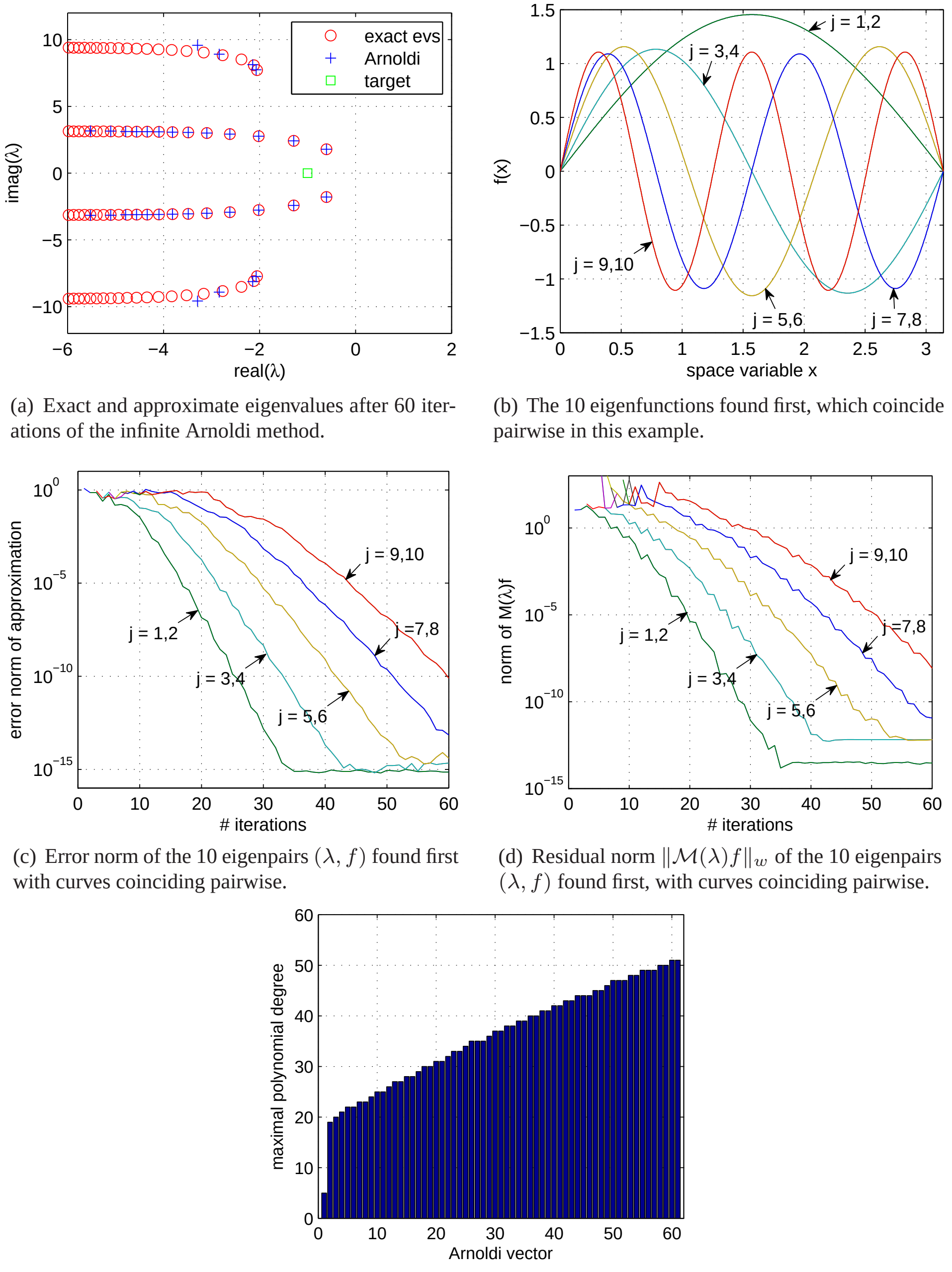
$$-j^2 = \lambda + e^{-\tau\lambda}.$$

It is straightforward to solve this equation for λ by using the Lambert W-function [10]. We find that the eigenvalues of the nonlinear operator eigenvalue problem are given by

$$\lambda = -j^2 + \frac{1}{\tau} W_\ell(-\tau e^{\tau j^2})$$

for any $j \in \mathbb{N}_+$ and any $\ell \in \mathbb{Z}$ where W_ℓ is the ℓ-th branch of the Lambert W-function. Note that different eigenvalues can have the same eigenfunction as the eigenfunctions do not depend on ℓ. The exact eigenvalues are shown in Figure 4.1(a). For our infinite Arnoldi procedure we have chosen the target $\mu = -1$, and the starting vector φ_1 was a polynomial of degree 5 with random (normally distributed) coefficients in the Chebyshev basis. Figure 4.1(a) also displays the approximate eigenvalues after 60 iterations of the infinite Arnoldi method, and Figure 4.1(b) displays the 10 approximate eigenfunctions f to which this method converged first. (Each two if these eigenfunctions coincide.)

The error norm for each of the 10 approximate eigenfunctions compared to the exact solution as a function of the number of Arnoldi iterations is shown in Figure 4.1(c) (there are always two error curves overlaying each other). Our spatial discretization was adapted such that the expected truncation error in the Chebyshev expansion is of the order of machine precision. We observe an error decay for each eigenfunction to about the same accuracy level as the number of Arnoldi iterations increases. The residual norm $\|\mathcal{M}(\lambda)f\|_w$ for each of the 10 approximate eigenpairs (λ, f) is shown in Figure 4.1(d) as a function of the number of Arnoldi iterations. Note how the degrees of Arnoldi vectors grow moderately with each Arnoldi iteration as depicted in Figure 4.1(e). More precisely, we display here the maximal degree among all polynomials collected in each block Arnoldi vector. This growth is expected because we potentially discover approximations to increasingly “nonsmooth” eigenvectors (i.e., those which are difficult to approximate by polynomials of low degree).



(a) Exact and approximate eigenvalues after 60 iterations of the infinite Arnoldi method.

(b) The 10 eigenfunctions found first, which coincide pairwise in this example.

(c) Error norm of the 10 eigenpairs (λ, f) found first with curves coinciding pairwise.

(d) Residual norm $\|\mathcal{M}(\lambda)f\|_w$ of the 10 eigenpairs (λ, f) found first, with curves coinciding pairwise.

(e) Polynomial degree of the Arnoldi vectors.

FIG. 4.1. *A differential equation with time delay (Section 4.1).*

4.2. Vibrating string with boundary control. We now consider a vibrating string on an interval $[0, L]$ with a clamped boundary condition at $x = 0$ and a feedback law at $x = L$. The feedback law is constructed with the goal to damp the vibrations of the string. In practice, a feedback control may only be available at a later point in time due to, e.g., a delay in measurement or the time required for calculating the feedback parameters. In such a situation the vibrating string is governed by a PDE with delay for $u : [0, L] \times [-\tau, \infty) \to \mathbb{R}$,

$$u_{tt}(x,t) = c^2 u_{xx}(x,t), \tag{4.3a}$$

$$u(0,t) = 0, \tag{4.3b}$$

$$u_x(L,t) = \alpha u_t(L, t-\tau), \tag{4.3c}$$

where c is the wave speed, τ is the delay, and α corresponds to a feedback law. See [15, 33] and the references therein for PDEs with delays and in particular the wave equation. In our setting, the eigenvalues associated with (4.3) are described by the λ-dependent operator

$$\mathcal{M}(\lambda) = \lambda^2 \mathcal{I} - c^2 \frac{\partial^2}{\partial x^2},$$

with λ-dependent boundary conditions,

$$\begin{aligned} c_1(\lambda, f) &= f(0) \\ c_2(\lambda, f) &= f'(L) - \alpha\lambda e^{-\tau\lambda} f(L). \end{aligned}$$

We now provide the implementation details for this example by specifying how to set up the differential equation (2.14). First note that

$$\begin{aligned} \mathcal{M}^{(1)}(\mu) &= 2\mu\mathcal{I}, \\ \mathcal{M}^{(2)}(\mu) &= 2\mathcal{I}. \end{aligned}$$

In our algorithm we require the derivatives of the boundary condition with respect to λ, which are explicitly given for $k > 0$ by

$$\begin{aligned} c_1^{(k)}(\mu, f) &= 0, \\ c_2^{(k)}(\mu, f) &= -\alpha f(L) e^{-\tau\mu} (-\tau)^{k-1} (k - \tau\mu). \end{aligned}$$

Hence, the specialization of (2.14) to this example is, for $p = k > 1$,

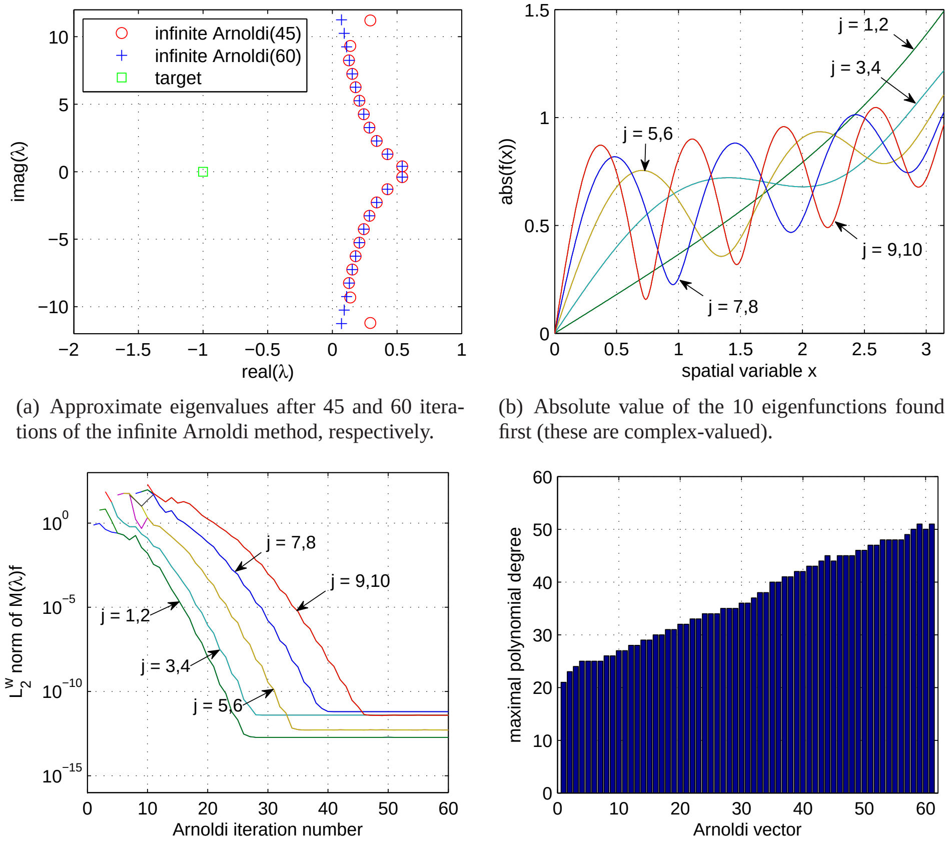
$$\mu^2 \varphi_1(x) - c^2 \varphi_1''(x) = -2\mu\psi_1(x) - \frac{1}{2} 2\psi_2(x) \tag{4.4a}$$

$$\varphi_1(0) = 0 \tag{4.4b}$$

$$\varphi_1'(L) - \alpha\mu e^{-\tau\mu} \varphi_1(L) = \alpha e^{-\tau\mu} \left(\sum_{j=1}^{k} \frac{1}{j} \psi_j(L) (-\tau)^{j-1} (j - \tau\mu) \right), \tag{4.4c}$$

where the functions $\psi_1, \ldots, \psi_k$ are given and $\varphi_1 \in L_1([a, b])$ has to be computed. When $p = k = 1$, i.e., in the first iteration, the term ψ_2 should be set to zero in the inhomogeneous term in (4.4a), whereas (4.4b) and (4.4c) remain the same for $p = k = 1$. Note that (4.4) is just a second order inhomogeneous differential equation with one Dirichlet and one Robin boundary condition.

In Figure 4.2 we visualize the computed approximate eigenvalues and (complex) eigenvectors of $\mathcal{M}$, as well as the decay of the residual norms $\|\mathcal{M}(\lambda) f\|_w$ for the first 10 approximate eigenpairs with λ closest to the target $\mu = -1$. The involved constants have been chosen as $\alpha = 1$, $c = 1$, and $\tau = 0.1$. The infinite Arnoldi method performs well on this example (for which an analytical solution does not seem to be available): after about 45 iterations the first 10 eigenpairs (λ, f) are resolved nicely while the degree of the Arnoldi vectors grows moderately.



(a) Approximate eigenvalues after 45 and 60 iterations of the infinite Arnoldi method, respectively.

(b) Absolute value of the 10 eigenfunctions found first (these are complex-valued).

(c) Residual norm $\|\mathcal{M}(\lambda)f\|_w$ of the 10 eigenpairs (λ, f) found first, with curves coinciding pairwise.

(d) Polynomial degree of the Arnoldi vectors.

FIG. 4.2. *Vibrating string with boundary control (Section 4.2.)*

4.3. An artificial example. In order to illustrate the broad applicability of our method we will now consider the following artificial nonlinear operator eigenvalue problem, which is complex, involves coefficient functions with branch cuts and a non-local operator in space. We use an interval $[a, b] = [0, \pi]$ and an operator defined by

$$\mathcal{M}(\lambda) = \frac{\partial^2}{\partial x^2} + \lambda\mathcal{I} + i(\lambda - \sigma_1)^{1/2}\mathcal{R} + i(\lambda - \sigma_2)^{1/2}\sin(x)\frac{\partial}{\partial x}$$

with boundary conditions

$$c_1(\lambda, f) = f(0)$$
$$c_2(\lambda, f) = \lambda f(\pi) - f'(\pi).$$

Here $\mathcal{R}$ represents the reversal operator $\mathcal{R} : u(x) \mapsto u(\pi - x)$. We let $\sigma_1 = -5$, $\sigma_2 = -10$ and let $\mu = 15$ be the target, so that the algorithm is expected to eventually find all eigenvalues in the disk $\mathbb{D}(15, 20)$.

The derivatives of the operator with respect to λ are given by

$$\begin{aligned}\mathcal{M}^{(1)}(\lambda) &= \mathcal{I} + i\frac{1}{2}(\lambda-\sigma_1)^{-1/2}\mathcal{R} + i\frac{1}{2}(\lambda-\sigma_2)^{-1/2}\sin(x)\frac{\partial}{\partial x},\\ \mathcal{M}^{(k)}(\lambda) &= -i(-2)^{-k}\left(1\cdot 3\cdot 5\cdots(2k-3)\right)(\lambda-\sigma_1)^{-(2k-1)/2}\mathcal{R}\\ &\qquad - i(-2)^{-k}\left(1\cdot 3\cdot 5\cdots(2k-3)\right)(\lambda-\sigma_2)^{-(2k-1)/2}\sin(x)\frac{\partial}{\partial x},\quad k>1,\end{aligned}$$

and the derivatives of the boundary conditions are simply $c_1^{(k)}(\lambda,f)=0$, $k\geq 1$ and $c_2^{(1)}(\lambda,f)=f(\pi)$, $c_2^{(k)}(\lambda,f)=0$ for $k\geq 2$.

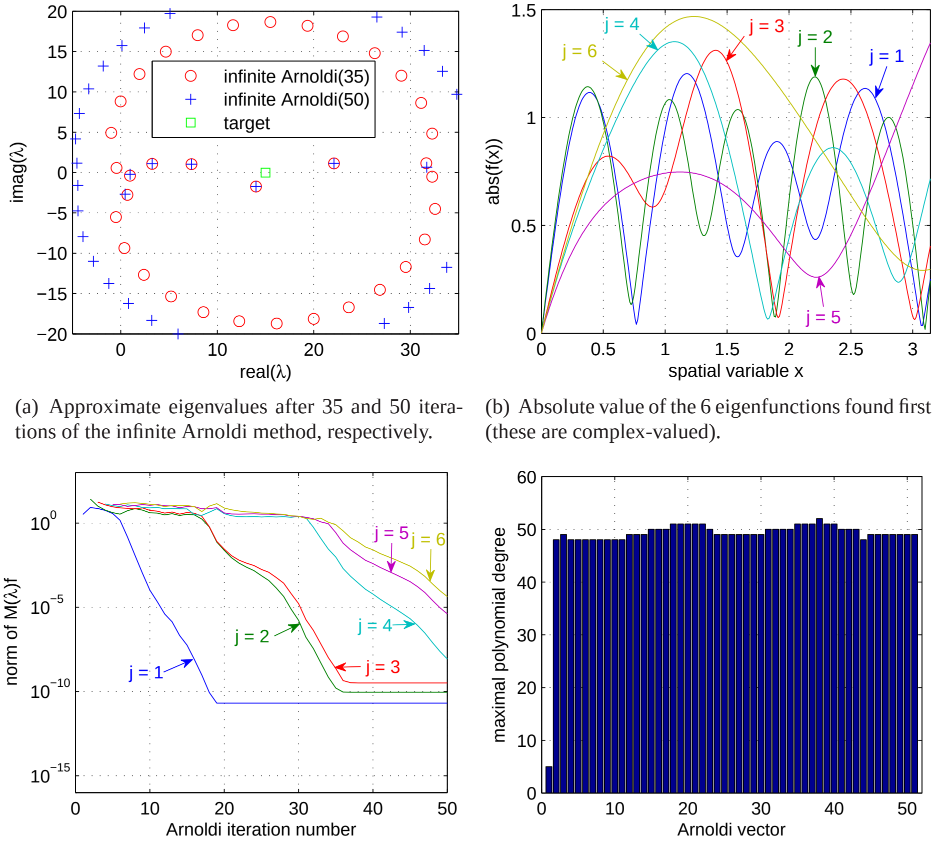
The numerical results are illustrated in Figure 4.3. Although the Arnoldi method still performs robustly, convergence is somewhat slower than for the previous two examples (see Figure 4.3(c)). A possible explanation may be given by the fact that the eigenvectors f of this problem have singularities nearby the interval $[a,b]$ (see how the polynomial degree of the Arnoldi vectors shown in Figure 4.3(d) increases to about 48 immediately after the first iteration), and therefore the Arnoldi method requires more iterations to resolve these.

A beautiful observation from Figure 4.3(a) is that the Arnoldi method starts to find spurious eigenvalues near the boundary of the disk of convergence $\mathbb{D}(15,20)$. (For iteration numbers higher than 50 this effect becomes even more pronounced.) This phenomenon is possibly related to a classical result from approximation theory due to Jentzsch [20], which states that the zeros of a truncated Taylor series have limit points everywhere on the boundary of the disk of convergence $\mathbb{D}(\mu,r)$. Note that all our theoretical results are valid only inside $\mathbb{D}(\mu,r)$, so that the appearance of spurious eigenvalues outside this set is not a contradiction of the theory. Of course these spurious eigenvalues will have large residuals associated with them, so that they are easily detectable even if the radius of convergence $r=20$ is unknown. A more detailed investigation of the convergence behavior of the infinite Arnoldi method and the interesting phenomenon of spurious eigenvalues will be subject of future work.

5. Concluding remarks and outlook.

A key contribution of this paper is the formulation of an Arnoldi-type iteration for solving nonlinear operator eigenproblems. Our approach relies on a dynamic representation of the Krylov vectors in the infinite Arnoldi algorithm, which are resolved automatically such that their trailing Chebyshev coefficients are of the order of machine precision and with the aim to compute eigenpairs to very high precision. It would be interesting to see if the spectral method recently proposed in [27] could further improve the accuracy of solutions computed with our algorithm. We have focused on the situation where the functions are of the type $f:[a,b]\to\mathbb{C}$, mostly, but not entirely, for notational convenience. The abstract formulation of the algorithm in Section 2 carries over to higher dimensions, e.g., to functions $f:\mathbb{R}^2\to\mathbb{C}$. However, in higher dimensions, the automatic adaption of the spatial resolution advocated in Section 3 becomes more delicate. A suitable function representation for two-dimensional problems highly depends on the geometry of the domain and is outside the scope of this paper. For PDEs with complicated geometries, the finite-element method (FEM) is a popular approach to representing functions. One could, of course, represent functions on such geometries using a (high-order) finite-element basis and carry out Algorithm 1, but it is not clear whether such a FEM-based infinite Arnoldi variant of Algorithm 1 would be computationally feasible (because it requires the solution of a PDE at each iteration).

The treatment of boundary conditions in the presented algorithm is, to our knowledge, somewhat novel and attractive. Note that boundary conditions nonlinear in λ can be handled in a general fashion, and their effect is simply propagated into the differential equation (2.14), i.e., the equation to be solved at every iteration. Some boundary conditions being nonlinear



(a) Approximate eigenvalues after 35 and 50 iterations of the infinite Arnoldi method, respectively.

(b) Absolute value of the 6 eigenfunctions found first (these are complex-valued).

(c) Residual norm $\|\mathcal{M}(\lambda)f\|_w$ of the 6 eigenpairs (λ, f) found first.

(d) Polynomial degree of the Arnoldi vectors.

FIG. 4.3. *The artificial example (Section 4.3).*

in λ can also be treated in a discretize-first approach, e.g., the derivative could be estimated by one-sided finite differences. We are, however, not aware of a generic procedure to incorporate nonlinear boundary conditions in a discretize-first approach.

We wish to point out that in [19], two variants of the infinite Arnoldi method are presented, and here we worked out the "Taylor version" of this method. An adaption of the presented algorithm along the lines of the "Chebyshev version" appears feasible although a completely different reasoning might be needed. We believe that our approach of dynamic representations of the Krylov vectors can be combined with the NLEIGS method presented in [16], which is based on rational interpolation instead of polynomial expansions. Besides these extensions, there are also several theoretical challenges that we wish to investigate in future work. For example, it would be interesting to understand how our special choice of the starting vector influences the convergence of the Arnoldi method and to characterize in which cases breakdown may appear and how it could be detected and handled.

Acknowledgments. The first author gratefully acknowledges the support of a Dahlquist research fellowship. We also acknowledge the valuable discussions with Prof. Wim Michiels, KU Leuven. We have gained much inspiration from the concepts of the software package Chebfun [3], and the authors acknowledge valuable discussions with Alex Townsend, Nick

Trefethen, and the rest of the Chebfun team. We would also like to thank the two anonymous referees whose insightful comments have considerably improved our paper.

REFERENCES

[1] J. Appell, E. De Pascale, and A. Vignoli, *Nonlinear Spectral Theory*, de Gruyter, Berlin, 2004.

[2] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura, *A numerical method for nonlinear eigenvalue problems using contour integrals*, JSIAM Letters, 1 (2009), pp. 52–55.

[3] Z. Battles and L. N. Trefethen, *An extension of MATLAB to continuous functions and operators*, SIAM J. Sci. Comput., 25 (2004), pp. 1743–1770.

[4] W.-J. Beyn, *An integral method for solving nonlinear eigenvalue problems*, Linear Algebra Appl., 436 (2012), pp. 3839–3863.

[5] J. P. Boyd, *Chebyshev and Fourier spectral methods*, 2nd. ed., Dover Publications, Mineola, NY, 2001.

[6] D. Breda, S. Maset, and R. Vermiglio, *Pseudospectral approximation of eigenvalues of derivative operators with non-local boundary conditions*, Appl. Numer. Math., 56 (2006), pp. 318–331.

[7] ———, *Numerical approximation of characteristic values of partial retarded functional differential equations*, Numer. Math., 113 (2009), pp. 181–242.

[8] ———, *Computation of asymptotic stability for a class of partial differential equations with delay*, J. Vib. Control, 16 (2010), pp. 1005–1022.

[9] F. Chatelin, *Spectral Approximation of Linear Operators*, Academic Press, New York, 1983.

[10] R. Corless, G. Connet, D. Hare, D. J. Jeffrey, and D. Knuth, *On the Lambert W function*, Adv. Comput. Math., 5 (1996), pp. 329–359.

[11] T. A. Driscoll, *Automatic spectral collocation for integral, integro-differential, and integrally reformulated differential equations*, J. Comput. Phys., 229 (2010), pp. 5980–5998.

[12] T. A. Driscoll, F. Bornemann, and L. N. Trefethen, *The chebop system for automatic solution of differential equations*, BIT, 48 (2008), pp. 701–723.

[13] C. Effenberger and D. Kressner, *Chebyshev interpolation for nonlinear eigenvalue problems*, BIT, 52 (2012), pp. 933–951.

[14] L. Fox and I. B. Parker, *Chebyshev Polynomials in Numerical Analysis*, Oxford University Press, London, 1968.

[15] M. Gugat, *Boundary feedback stabilization by time delay for one-dimensional wave equations*, IMA J. Math. Control Inf., 27 (2010), pp. 189–203.

[16] S. Güttel, R. V. Beeumen, K. Meerbergen, and W. Michiels, *NLEIGS: A class of robust fully rational Krylov methods for nonlinear eigenvalue problems*, Tech. Report, MIMS EPrint 2013.49, Manchester Institute for Mathematical Science, University of Manchester, 2013.

[17] A. C. Hansen, *Infinite-dimensional numerical linear algebra: theory and applications*, Proc. R. Soc. Lond., Ser. A, Math. Phys. Eng. Sci., 466 (2010), pp. 3539–3559.

[18] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-dependent Problems*, Cambridge University Press, Cambridge, 2007.

[19] E. Jarlebring, W. Michiels, and K. Meerbergen, *A linear eigenvalue algorithm for the nonlinear eigenvalue problem*, Numer. Math., 122 (2012), pp. 169–195.

[20] R. Jentzsch, *Untersuchungen zur Theorie der Folgen analytischer Funktionen*, Acta Math., 41 (1916), pp. 219–251.

[21] A. B. Kuijlaars, *Convergence analysis of Krylov subspace iterations with methods from potential theory*, SIAM Rev., 48 (2006), pp. 3–40.

[22] C. Lanczos, *Trigonometric interpolation of empirical and analytical functions*, J. Math. Phys. Mass. Inst. Tech., 17 (1938), pp. 123–199.

[23] S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Structured polynomial eigenvalue problems: good vibrations from good linearizations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1029–1051.

[24] ———, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 971–1004.

[25] V. Mehrmann and H. Voss, *Nonlinear eigenvalue problems: a challenge for modern eigenvalue methods*, GAMM Mitt., 27 (2004), pp. 121–152.

[26] S. Olver, *Shifted GMRES for oscillatory integrals*, Numer. Math., 114 (2010), pp. 607–628.

[27] S. Olver and A. Townsend, *A fast and well-conditioned spectral method*, SIAM Rev., 55 (2013), pp. 462–489.

[28] Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.

[29] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

[30] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, Philadelphia, 2000.

[31] ———, *Approximation Theory and Approximation Practice*, SIAM, Philadelphia, 2013.

[32] H. VOSS, *Nonlinear eigenvalue problems*, in Handbook of Linear Algebra, L. Hogben, ed., Taylor & Francis, Boca Raton, Fl, 2014, pp. 60-1–60-24.

[33] J. WU, *Theory and Applications of Partial Functional-Differential Equations*, Springer, NY, 1996.